

# USE CASE

\*red colour font means those are not implemented

## Components required-

1. **Raspberry pi 4b**
2. **ESP8266 (NodeMCU)**
3. **Relay Switch**
4. **USB data cable**
5. **Jumper wires**

## **OS NEEDED**

**Raspberry Pi OS with desktop**

**Release date: March 4th 2021**

**Kernel version: 5.10**

**Size: 1,175MB**

**SD card**

**16GB/32GB**

**Preferable high speed sdcard**

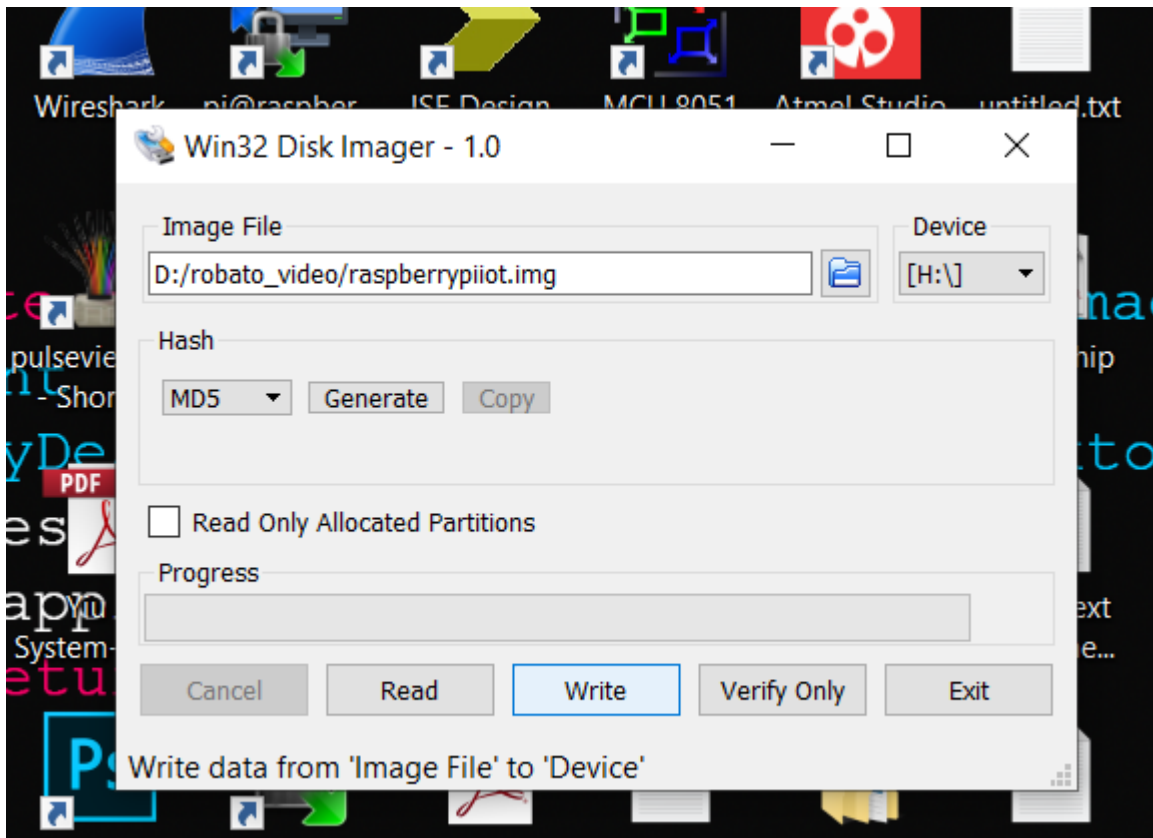
**Download the my os image (6gb compressed file)**

**<https://drive.google.com/file/d/1XpPOEP87ycJtb-3VWN1jG7g3xcPNyEvy/view?usp=sharing>**

- **After extracting the zipfile**
- **then download**
- **disk32 imager to install os to sdcard**

<https://sourceforge.net/projects/win32diskimager/>

- **insert sdcard on pc using sdcard reader,select the location of the extracted os image**
- **raspberrypiot.img**
- **select MD5**
- **click on write**



# **1.DATABASE SETUP**

- **Check the videolink and refer doc for reference**
- **Watch all the videos in 720p or 1080p only**

<https://drive.google.com/file/d/1ioGUwRZF52IBJbPO1zuxdjH4e9j2qS4f/view?usp=sharing>

- **Checking the SQL database**
- **Already installed with the os.img**
- **Type cmd**

```
sudo mysql -u root -p
```

- **Type password**

```
password
```

## • **Database name**

- **SQL\_GATEWAY\_DATABASE**

## • **4 Tables**

**1. USERS\_DATA -(USERNAME,PASSWORD,SECURITYKEY)**

**2. GATEWAY\_CONFIG-(SSID,PASSWORD,GATEWAYIP)**

**3. ROOM\_CONTROLBOX\_APPLIANCE-**

**( DATE ,LOCATION ,CLIENT    APPLIANCE , ON\_TIME , OFF\_TIME ,  
TOTAL\_ON\_TIME )**

**4. CONTROLBOX\_CONFIG**

**(ROOM ,DEVICE ,APPLIANCE1,APPLIANCE2, APPLIANCE3,APPLIANCE4)**

- **TYPE CMD**

```
USE SQL_GATEWAY_DATABASE;
```

- **TYPE CMDS**

```
SELECT * FROM USERS_DATA;
```

```
SELECT * FROM GATEWAY_CONFIG;
```

```
SELECT * FROM ROOM_CONTROLBOX_APPLIANCE;
```

```
SELECT * FROM CONTROLBOX_CONFIG;
```

- **NEXT DELETE THE PREVIOUS DATAS FROM TABLE**

```
DELETE FROM USERS_DATA;
```

```
DELETE FROM GATEWAY_CONFIG;
```

```
DELETE FROM ROOM_CONTROLBOX_APPLIANCE;
```

```
DELETE FROM CONTROLBOX_CONFIG;
```

- **CHANGE THE ROOT PASSWORD AND SET YOUR OWN PASSWORD FOR SQL DATABASE**

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('MY_NEW_PASSWORD');
```

```
FLUSH PRIVILEGES;
```

**First customer will buy the product now below process is to be followed-**

- **Mobile – Gateway**

**Set Wi-Fi credentials via Bluetooth (Secure Login) – First time default fixed password.**

**Step 1 - Customer will install app in his smartphone to configure and control devices.**

**Step 2-** Customer will plug and turn ON the gateway (raspberry pi). When gateway starts, it will open its Bluetooth, now customer will connect the Bluetooth of gateway using a default password known to customer, and go to the installed app and configure (Wi-Fi credential) the gateway.

**Step 3-** Now gateway will connect with the wi-fi and obtain an ip address, and send the ip address to app using the Bluetooth.

**Step 4-** A python program will run a web server which will be accessed by the user app to configure the gateway in next step. Web page will be secured with a default id and password, which will be known to user.

- **BLUETOOTH CONFIGURATIONS TEST**

**VIDEO LINK-**

[https://drive.google.com/file/d/1Web\\_4l0AD6lAzoAecvCfiAdqYdt\\_Qw1i/view?usp=sharing](https://drive.google.com/file/d/1Web_4l0AD6lAzoAecvCfiAdqYdt_Qw1i/view?usp=sharing)

- **Bluetooth app1 link**

[https://res.cloudinary.com/di2vaxvhl/raw/upload/v1571090089/apk\\_Ubidots\\_source\\_code.zip](https://res.cloudinary.com/di2vaxvhl/raw/upload/v1571090089/apk_Ubidots_source_code.zip)

- **Procedure**

- **.Open the file containing run\_main.py**

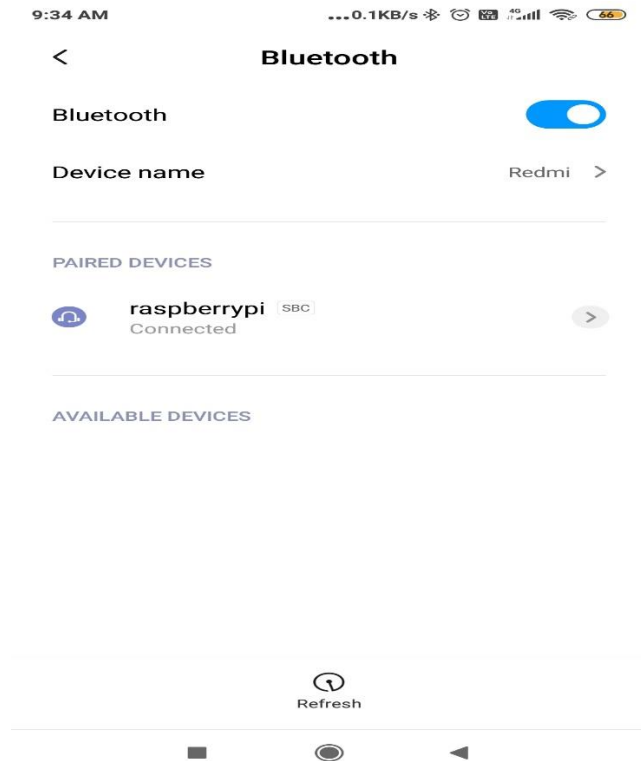
- **Type cmd**

cd bluetoothconfig

sudo python3 run.py

```
pi@raspberrypi:~ $ cd bluetoothconfig
pi@raspberrypi:~/bluetoothconfig $ sudo python3 run_main.py
wifi password configurations starting
waiting for connection on RFCOMM channel 1
```

- Pairing of Bluetooth with raspberry pi must have completed early before ,now just connect mobile Bluetooth to raspi bluetooth



Open the app type your wifi ssid and password, from your phones portable hotspot settings or if you are using Router accordingly type it with out mistake

9:35 AM

...1.5KB/s      

Paired Bluetooth Devices

raspberrypi

DC:A6:32:8B:CB:1B



REFRESH DEVICES

SSID

stijo

PSK

stijoseph

START CONFIGURATION

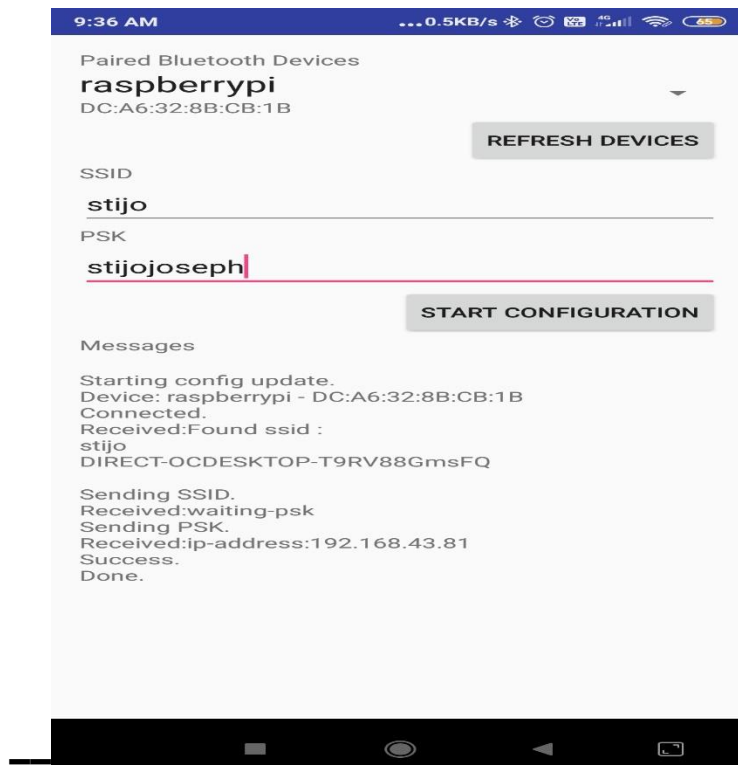
Messages



## OUTPUT ON RASPI

```
pi@raspberrypi:~ $ cd bluetoothconfig
pi@raspberrypi:~/bluetoothconfig $ sudo python3 run_main.py
wifi password configurations starting
Waiting for connection on RFCOMM channel 1
Accepted connection from ('D8:32:E3:45:4A:E1', 1)
Waiting for SSID...
ssid received
stijo
Waiting for PSK...
psk received
stijoseph
the recieved SSID matches with the discovered SSID
*****wait for 10 seconds please*****
OK
█
```

- AFTER WAITING FOR 10 SECONDS
- YOU WILL GET THE IP ADDRESS ON YOUR MOBILE





```

Waiting for connection on RFCOMM channel 1
Accepted connection from ('D8:32:E3:45:4A:E1', 1)
Waiting for SSID...
ssid received
stijo
Waiting for PSK...
psk received
stijoseph
the recieved SSID matches with the discovered SSID
*****wait for 10 seconds please*****
OK
wlan0 IEEE 802.11 ESSID:"stijo"
Mode:Managed Frequency:2.462 GHz Access Point: D8:32:E3:45:4A:E2
Bit Rate=65 Mb/s Tx-Power=31 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:on
Link Quality=70/70 Signal level=-37 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.43.81 netmask 255.255.255.0 broadcast 192.168.43.255
inet6 fe80::cd73:cd1b:f6b0:e880 prefixlen 64 scopeid 0x20<link>
inet6 2409:4073:4e0b:8f2a:b0f7:8354:53ec:fe3f prefixlen 64 scopeid 0x0<global>
ether dc:a6:32:8b:cb:1a txqueuelen 1000 (Ethernet)
RX packets 249 bytes 36228 (35.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 140 bytes 18712 (18.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

data send to sql database
stijo stijoseph 192.168.43.81
Waiting for connection on RFCOMM channel 1
#####

Again Waiting for connection on RFCOMM channel 1 go to another serial monitor bluetooth app and connect again from mobile

Enter MSG(json format) as per the given format in document
#####

```

**CONTINUATION TO THIS PROGRAM SENDING**

**USERNAME,PASSWORD,SECURITY IN JSON FORMAT IS IN  
NEXT PAGE**

**IF YOU SEE SIMILAR KIND OF OUTPUT IT ,MEANS THE DATA IS  
SENT TO THE SQL DATABASE,THE IP ADDRESS,SSID  
,PASSWORD ARE STORED IN THE GATEWAY\_CONFIG TABLE  
AND THIS IPADDRESS WILL BE USED AS MQTTBROKER  
ADDRESS FOR SUBSCRIBING AND PUBLISHING FOR THE REST  
OF THE PROGRAMS  
CHECK THE DATABASE FOR CONFIRMATION**

```

pi@raspberrypi:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 17
Server version: 10.3.27-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE SQL_GATEWAY_DATABASE;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [SQL_GATEWAY_DATABASE]> SELECT * FROM GATEWAY_CONFIG;
+-----+-----+-----+
| SSID | PASSWORD | IP_ADDRESS |
+-----+-----+-----+
| stijo | stijoseph | 192.168.43.81 |
+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [SQL_GATEWAY_DATABASE]> █

```

## Set Gateway settings via WIFI server (Secure Login) – First time default fixed password.

**Step 1- Now app knows the ip address of raspberry pi, user will turn off the mobile Bluetooth and switch to wi-fi. App will automatically connect with the ip of gateway.**

**Step 2- Now app will request for the http webpage so that user can configure the gateway for mqtt. Python running web server will response with a web page.**

**Step 3- Webpage will be secured with a default id and password, customer will enter the default id and password and if it is correct webpage will show a option to change the customer's id, password and security key.**

After the mobile gateway setting are over ,an another Bluetooth channel will start

Here the userid ,password,securitykey,command is sent via Bluetooth itself.For this I use another Bluetooth app with Serial monitor

In the mobile app type in the serial monitor

App2 download link

[https://play.google.com/store/apps/details?id=de.kai\\_morich.serial\\_bluetooth\\_terminal](https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal)

If you want to reconfigure ssid and password after uploading this username,password,securitykey , put “wificonfig” as json value of command , Here the command value is to decide which mode the program should go into or else to add more username,password etc to sql keep the command value as “config”

Msg format:

```
{ “userid” : “your_username” , “password” : “your_password” , “securitykey” : “secure”, “command” : “wificonfig”}
```

//program restarts to configure ssid password and all afer saving this uername password etc

PREVIOUS PROGRAM IN RUNNING MODE

```

Waiting for connection on RFCOMM channel 1
Accepted connection from ('D8:32:E3:45:4A:E1', 1)
Waiting for SSID...
ssid received
stijo
Waiting for PSK...
psk received
stijojoseph
the recieved SSID matches with the discovered SSID
*****wait for 10 seconds please*****
OK
wlan0 IEEE 802.11 ESSID:"stijo"
Mode:Managed Frequency:2.462 GHz Access Point: D8:32:E3:45:4A:E2
Bit Rate=65 Mb/s Tx-Power=31 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:on
Link Quality=70/70 Signal level=-37 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.43.81 netmask 255.255.255.0 broadcast 192.168.43.255
inet6 fe80::cd73:cd1b:f6b0:e880 prefixlen 64 scopeid 0x20<link>
inet6 2409:4073:4e0b:8f2a:b0f7:8354:53ec:fe3f prefixlen 64 scopeid 0x0<global>
ether dc:a6:32:8b:cb:1a txqueuelen 1000 (Ethernet)
RX packets 249 bytes 36228 (35.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 140 bytes 18712 (18.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

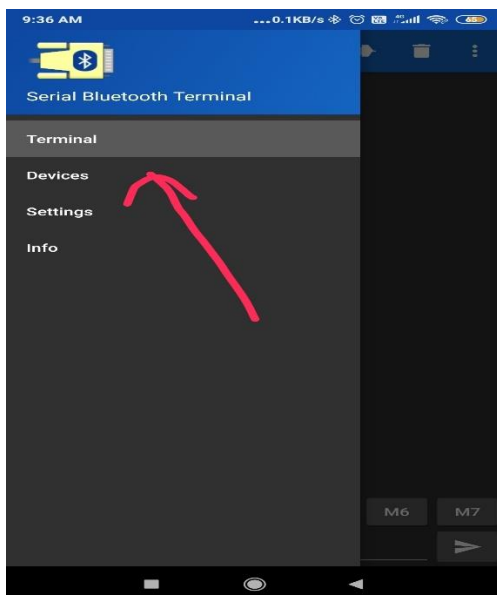
data send to sql database
stijo stijojoseph 192.168.43.81
Waiting for connection on RFCOMM channel 1
#####

Again Waiting for connection on RFCOMM channel 1 go to another serial monitor bluetooth app and connect again from mobile

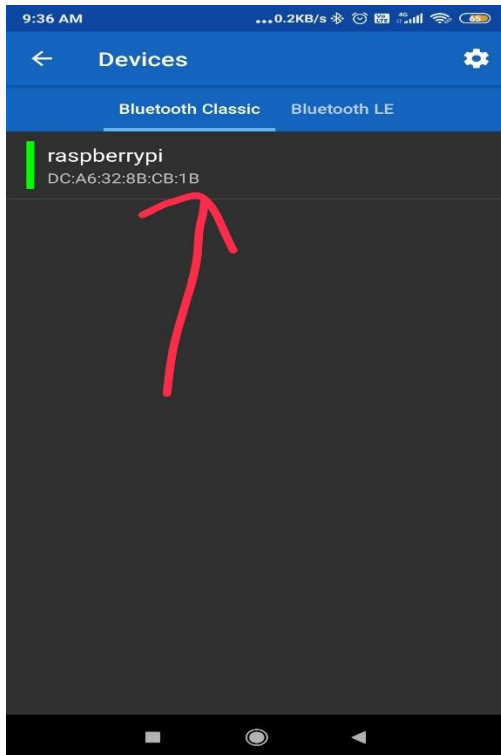
Enter MSG(json format) as per the given format in document
#####

```

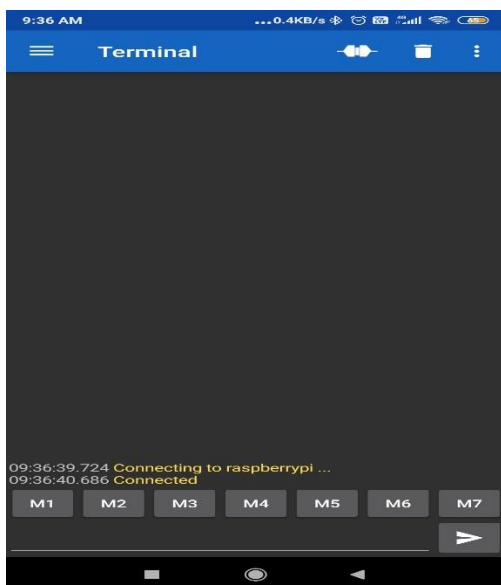
## TOUCH ON DEVICES



## SELECT RASPBERRYPI



**AFTER SELECTING RASPBERRY PI YOU CAN SEE CONNECTED STATUS ON MOBILE TERMINAL**



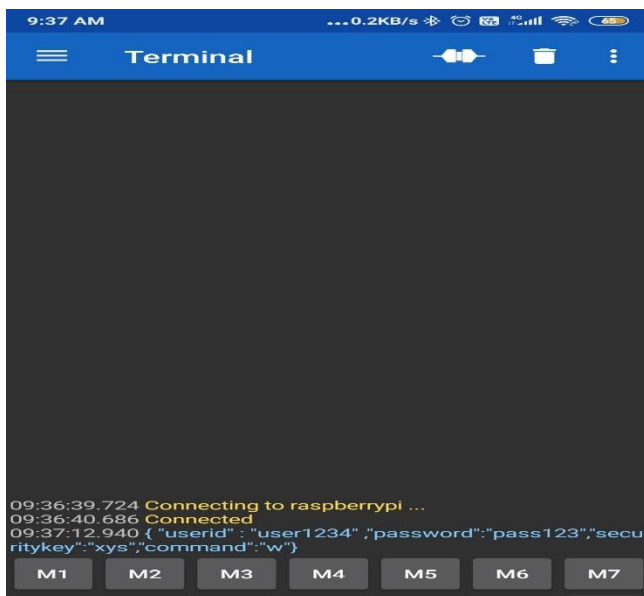
**Msg format:**

```
{ "userid" : "your_username" , "password" : "your_password" , "securitykey" :  
"secure", "command" : "config"}
```

In the command value if anyother value other than “wificonfig” is given the program will move into user details addition mode ,for wifi settings configure again u may add “wificonfig” as command value else its your wish

And touch send option

**After sending the terminal will look like this**



**Pi terminal will be like this**

```

stijo
Waiting for PSK...
psk received
stijojoseph
the recieved SSID matches with the discovered SSID
*****wait for 10 seconds please*****
OK
wlan0 IEEE 802.11 ESSID:"stijo"
Mode:Managed Frequency:2.462 GHz Access Point: D8:32:E3:45:4A:E2
Bit Rate=65 Mb/s Tx-Power=31 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:on
Link Quality=70/70 Signal level=-37 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.43.81 netmask 255.255.255.0 broadcast 192.168.43.255
inet6 fe80::cd73:cd1b:f6b0:e880 prefixlen 64 scopeid 0x20<link>
inet6 2409:4073:4e0b:8f2a:b0f7:8354:53ec:fe3f prefixlen 64 scopeid 0x0<global>
ether dc:a6:32:8b:cb:1a txqueuelen 1000 (Ethernet)
RX packets 249 bytes 36228 (35.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 140 bytes 18712 (18.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

data send to sql database
stijo stijojoseph 192.168.43.81
Waiting for connection on RFCOMM channel 1
#####

Again Waiting for connection on RFCOMM channel 1 go to another serial monitor bluetooth app and connect again from mobile

Enter MSG(json format) as per the given format in document
#####

Accepted connection from ('D8:32:E3:45:4A:E1', 1)
{ "userid" : "user1234" , "password": "pass123", "securitykey": "xys", "command": "w"}
new user added

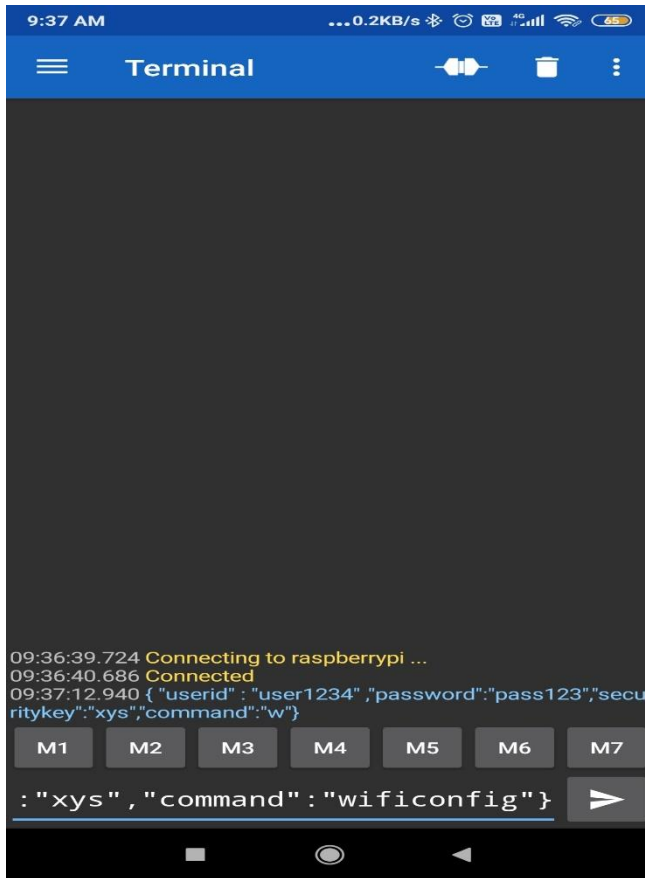
```

**//program restarts to add more username ,password,securitykey**

**Msg format**

**{ “userid” : “your\_username” , “password” : “your\_password” , “securitykey” : “secure”, “command” : “wificonfig”}**

**this msgformat is for adding user details and shift to wifi configuration mode**



## After sending

```
*****wait for 10 seconds please*****
OK
wlan0 IEEE 802.11 ESSID:"stijo"
Mode:Managed Frequency:2.462 GHz Access Point: D8:32:E3:45:4A:E2
Bit Rate=65 Mb/s Tx-Power=31 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:on
Link Quality=70/70 Signal level=-37 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.43.81 netmask 255.255.255.0 broadcast 192.168.43.255
inet6 fe80::cd73:cd1b:f6b0:e880 prefixlen 64 scopeid 0x20<link>
inet6 2409:4073:4e0b:8f2a:b0f7:8354:53ec:fe3f prefixlen 64 scopeid 0x0<global>
ether dc:a6:32:8b:cb:1a txqueuelen 1000 (Ethernet)
RX packets 249 bytes 36228 (35.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 140 bytes 18712 (18.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

data send to sql database
stijo stijojoseph 192.168.43.81
Waiting for connection on RFCOMM channel 1
#####

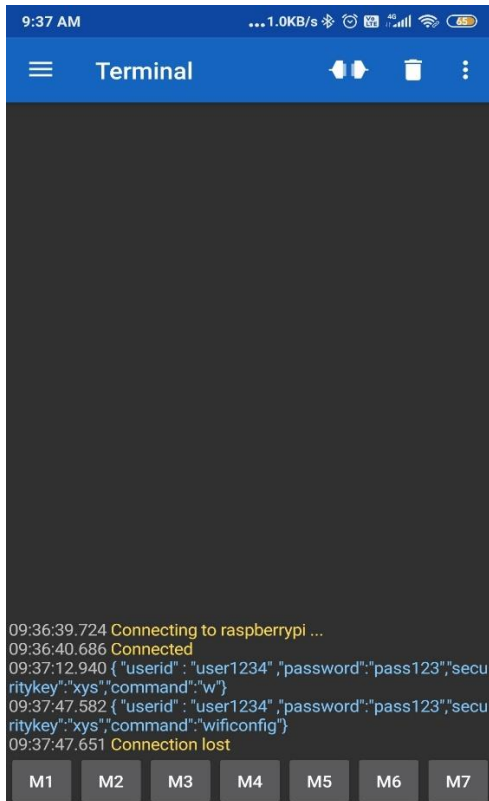
Again Waiting for connection on RFCOMM channel 1 go to another serial monitor bluetooth app and connect again from mobile

Enter MSG(json format) as per the given format in document
#####

Accepted connection from ('D8:32:E3:45:4A:E1', 1)
{ "userid" : "user1234", "password": "pass123", "securitykey": "xys", "command": "w" }
new user added

{ "userid" : "user1234", "password": "pass123", "securitykey": "xys", "command": "wificonfig" }
user1234
current user updated
wifi password configurations starting
Waiting for connection on RFCOMM channel 1
```





- **User details will be stored to database and connection will be lost because Bluetooth channel get restarted ,u need to use the other app for configuring wifi**
- **PRACTICALLY THIS PROGRAM SHOULD KEEP ON RUNNING FOR EVER IN BACKGROUND,MEANS ON BOOTING OF PI THIS run\_main.py SHOULD BE AUTOMATICALLY EXECUTED AND THE BLUETOOTH CHANNEL WILL BE OPEN EITHER FOR WIFICONFIGURATIONS OR FOR USER DETAILS ADDING,ANY NUMBER OF USERS CAN BE ADDED**



**Step 4- the customer's id and password will be saved as admin id and password in MySQL database under a table name: users\_data**

User id	Password	Security key
Admin id	Xyz	abcd

**Step 5- Now web server will response with a webpage to configure the following blocks-**

**Cloud**

**Server**

**URL End**

**client id**

**Cloud**

**Server**

**Topic**

**Location**

**Local/Both**

**(Local + Cloud)**

**Device Name**

**Access Key**

**Step 6- The web server will save these credential in the MySQL with table name gateway\_configuration.**

**Step 7- If admin user want to add more user, User will click on gateway setting in the app, again click on users in the option, gateway will response with a webpage with the following option:**



**Step 8- Now user will click on “Add user” option, again a window will open to enter username and password and security key and save it.**

**Step 9- The entered will be stored in the table in user\_data in MySQL database running on gateway.**

**\*\* The above steps are used to configure the gateway\*\***

## **3.MQTT SETUP**

VIDEO LINK

[https://drive.google.com/file/d/1RFYoZWvM\\_MEFN\\_t7rnT4lyY4zpa3AOyt/view?usp=sharing](https://drive.google.com/file/d/1RFYoZWvM_MEFN_t7rnT4lyY4zpa3AOyt/view?usp=sharing)

### **❖ Mqtt local setup**

#### **• ON RASPBERRY PI TERMINAL**

- Set an MQTT username and password
- Run this command to create an MQTT username and password (substitute YOUR-NEWMQTT-USERNAME):

```
sudo mosquitto_passwd -c /etc/mosquitto/pwfil YOUR-NEW-MQTT-USERNAME
```

- Now when you press Enter it will ask to create password.

.

```
sudo nano /etc/mosquitto/mosquitto.conf
```

- OPENING THIS WILL GIVE U A FILE, INSIDE CHANGE “pwfile” to “pwfil” .dont change or delete any other lines

```
pid_file /var/run/mosquitto.pid
persistence true
persistence_location /var/lib/mosquitto/
log_dest file /var/log/mosquitto/mosquitto.log
allow_anonymous false
password_file /etc/mosquitto/pwfil
```

click Ctrl+O > Enter > Ctrl+X to save the config file.

### Test the broker

Substitute YOUR-MQTT-USERNAME and YOUR-MQTT-PASSWORD with what you defined in the previous step

```
mosquitto_sub -d -u YOUR-MQTT-USERNAME -P YOUR-MQTT-PASSWORD -t dev/test
```

➤ You should get back a response similar to this:

Client mosqsub|730-hostname sending CONNECT

Client mosqsub|730-hostname received CONNACK (0)

Client mosqsub|730-hostname sending SUBSCRIBE (Mid: 1, Topic: dev/test, QoS: 0)

Client mosqsub|730-hostname received SUBACK

Press Ctrl-C to break out of it.

❖ . Setting

### SETTING UP THE CLOUD

#### Setting cloud mqtt broker

Visit <https://myqttHub.com/en>

- ☐ Enter your e-mail address > username (must be unique) > Create a password
- ☐ Now click on “Sign up with open plan”.
- ☐ Now it will send a verification mail to your registered email, go to your email and verify it. Verify email on the same PC/Laptop/pi where you have signed up for cloud broker.
- ☐ Now again visit <https://myqttHub.com/en> and login using your credential then go to home page.
- ☐ Click on MyqttHub (Top in the left), it will open a box, click on domains
- ☐ It will open a page MyQtt Domains, a domain name hub-username will be available there. Click on the domain name (hub-username).
- ☐ Now click on “Bootstrap”.
- ☐ Now click on “Add bootstrap Auth credentials”, create a username and password and save it. (Note it down, it will be used for client to connect with myqtt hub).

## . Setting mqtt bridge to connect local mqtt broker and cloud mqtt broker

- ❑ Before setting the mqtt bridge we have to stop running mosquitto broker-
- ❑ Open command prompt in raspberry pi and run the command  
`sudo service mosquitto stop`
- ❑ Now we will open mosquitto configure file to configure it as a mqtt bridge  
`sudo nano /etc/mosquitto/mosquitto.conf`
- ❑ Configure this file as given below. Most of the lines will be (Line started with '#' is comment). The words which are written in **Red colour**, have to be modified it accordingly.  
# Place your local configuration in /etc/mosquitto/conf.d/  
# A full description of the configuration file is at  
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

```
pid_file /var/run/mosquitto.pid
persistence true
persistence_location /var/lib/mosquitto/
log_dest file /var/log/mosquitto/mosquitto.log
allow_anonymous false
password_file /etc/mosquitto/pwfiles
#port for local mqtt broker
listener 1884
# connection name
connection node02
# address and port of broker with which you want to create a bridge. Here
#myqtthub
address node02.myqtthub.com:1883
cleansession true
keepalive_interval 60
notifications false
# start automatically after 30 sec. when pi starts.
start_type automatic
try_private true
# create a client id for the bridge
remote_clientid clientid
# enter password which you have created in bootstrap in myqtthub
remote_password password #put your boot strap cred password here ,from cloud
# enter username which you have created in bootstrap in myqtthub
remote_username username #put your boot strap cred username here, from cloud
# subscribe and publish all topic both sides.
```

topic # both

click Ctrl+O > Enter > Ctrl+X to save the config file.

Now reboot the pi (**sudo reboot**)

□ After rebooting open command prompt and enter

**sudo mosquitto -c mosquitto.conf -v**

□ Open another command prompt and run

**mosquitto -v**

Now mosquitto is running.

Download app

<https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod>

### **3.Mobile - Control Box**

**Set Control Box configuration – via control box hotspot  
connection (secure login)**

**Basic idea:**

**Inorder to make the esp8266/controlbox to Send the  
ssid,password,ipaddress, appliances names,Room name etc and rest  
all the configuration details we are going to send esp8266**

**By making it as a web server.**

**Means esp8266 upon uploading the code act as webserver**

**That is it starts a hotspot ,connect to the hotspot with mobile /pc**

**You need to configure the details in the webpage, and up on  
submission the datas .your details will be saved in its EEPROM**

**Config file.Next time restarting it gets connected to the wifi**

### **THE PROCEDURE PERFORMED BY USER**

**Step 1- User will plug and turn ON the control box.**

**Step 2- Control box will open its hotspot with a default Wi-Fi SSID and password.**

**Step 3- Customer will connect the smartphone wi-fi with the hotspot of control box  
and open the app and choose option to configure the control box.**

**Step 4- App will connect with control box with a default ip address.**



**Step 5- Control box will response a webpage to configure it.**

**Step 6- Now customer/app will configure the following blocks and save it –**

**Control Box Name**

**Room name (where it is to be  
implemented) Access Key**

**Location**

**Appliance 1**

**Appliance 2 (so on till Appliance 6, if  
available) PWM Appliance 1**

**PWM Appliance**

**2 Gateway ip,  
port Wi-Fi SSID**

**Wi-Fi pass**

**Step 7- The above configuration will also be saved in app and when app again connect with gateway it will sync the control box configuration with gateway using a webpage. Gateway will store the control box information in a table name control\_box\_configuration.**

**VIDEO LINK-**

**<https://drive.google.com/file/d/1Tqxda4HJ7a4wMc5zadwMRV1wgKHXAKHi/view?usp=sharing>**

**PROCEDURE NEEDED TO BE PERFORMED BY THE PROGRAMMER TO SETUP  
THE CONTROL BOX**

- 1.Connect the esp8266/controlbox to pc ,open the Arduino ide**
- 2.Compile the code ,before uploading the connect “D0” pin of esp8266 to its  
3V pin with jumper wire**

**There are two modes for this esp8266**

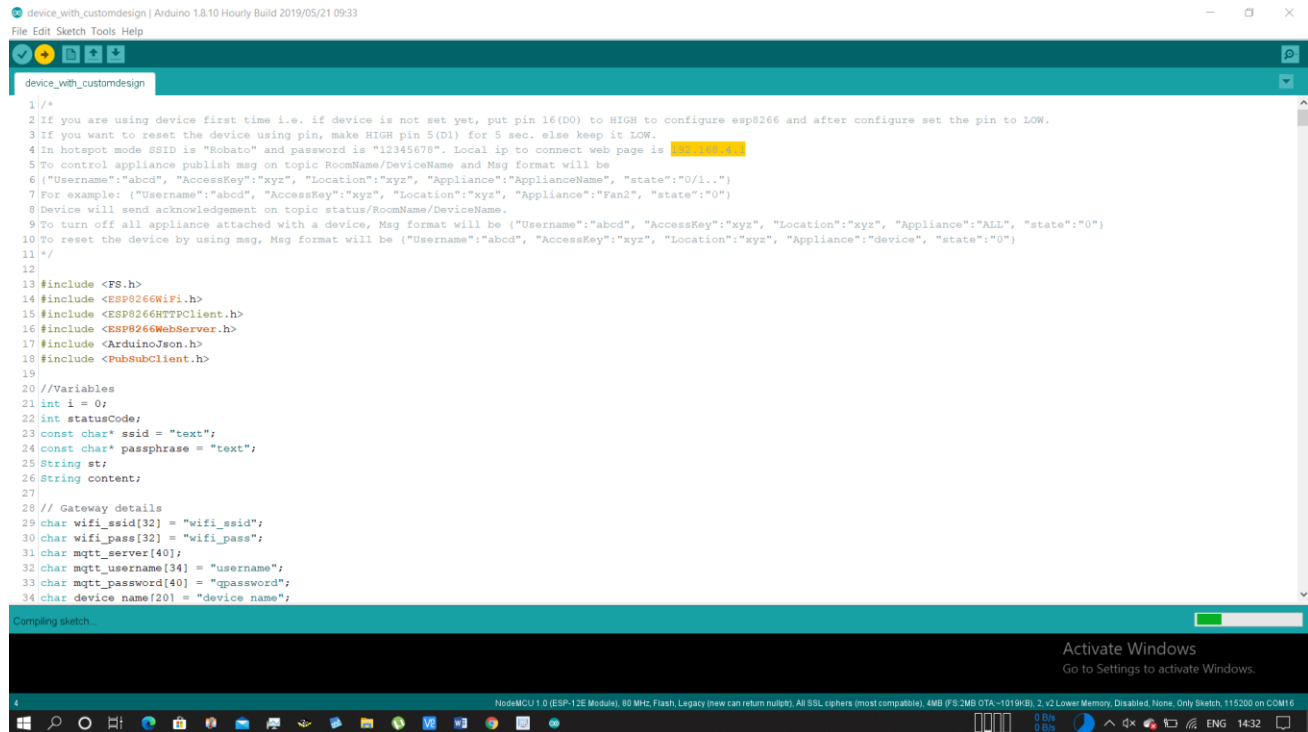
- a. Web server – to configure the details to work as control box**
- b. Control box – for appliance on/off and sending ack to gateway via mqtt**

**note:by connect D0 to 3V when are making it as webserver to configure details**

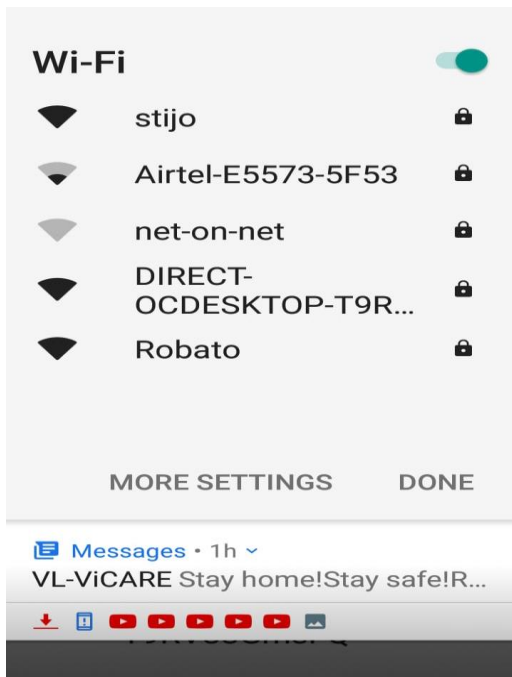
**here**

**upload the code and open the serial monitor to see the output**

**D0 should be connected to 3v**

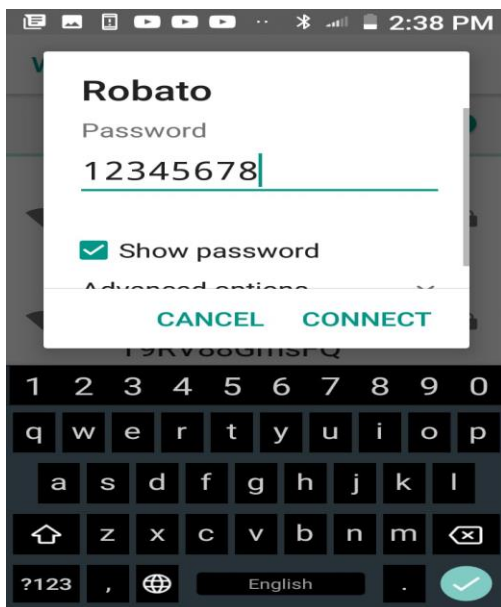


```
1 /*
2 3 If you are using device first time i.e. if device is not set yet, put pin 16(D0) to HIGH to configure esp8266 and after configure set the pin to LOW.
3 4 If you want to reset the device using pin, make HIGH pin 5(D1) for 5 sec. else keep it LOW.
4 5 In hotspot mode SSID is "Robato" and password is "12345678". Local ip to connect web page is 192.168.4.1
5 6 To control appliance publish msg on topic RoomName/DeviceName and Msg format will be
6 7 {"Username":"abcd", "AccessKey":"xyz", "Location":"xyz", "Appliance":"ApplianceName", "state":"0/1.."}
7 8 For example: {"Username":"abcd", "AccessKey":"xyz", "Location":"xyz", "Appliance":"Fan2", "state":"0"}
8 9 Device will send acknowledgement on topic status/RoomName/DeviceName.
9 10 To turn off all appliance attached with a device, Msg format will be {"Username":"abcd", "AccessKey":"xyz", "Location":"xyz", "Appliance":"ALL", "state":"0"}
10 11 To reset the device by using msg, Msg format will be {"Username":"abcd", "AccessKey":"xyz", "Location":"xyz", "Appliance":"device", "state":"0"}
11 12 */
12 13 #include <FS.h>
13 14 #include <ESP8266WiFi.h>
14 15 #include <ESP8266HTTPClient.h>
15 16 #include <ESP8266WebServer.h>
16 17 #include <ArduinoJson.h>
17 18 #include <PubSubClient.h>
18 19
19 20 //Variables
20 21 int i = 0;
21 22 int statusCode;
22 23 const char* ssid = "text";
23 24 const char* passphrase = "text";
24 25 String st;
25 26 String content;
26 27
27 28 // Gateway details
28 29 char wifi_ssid[32] = "wifi_ssid";
29 30 char wifi_pass[32] = "wifi_pass";
30 31 char mqtt_server[40];
31 32 char mqtt_username[34] = "username";
32 33 char mqtt_password[40] = "password";
33 34 char device_name[20] = "device name";
34 35
35 36
36 37
37 38
38 39
39 40
40 41
41 42
42 43
43 44
44 45
45 46
46 47
47 48
48 49
49 50
50 51
51 52
52 53
53 54
54 55
55 56
56 57
57 58
58 59
59 60
60 61
61 62
62 63
63 64
64 65
65 66
66 67
67 68
68 69
69 70
70 71
71 72
72 73
73 74
74 75
75 76
76 77
77 78
78 79
79 80
80 81
81 82
82 83
83 84
84 85
85 86
86 87
87 88
88 89
89 90
90 91
91 92
92 93
93 94
94 95
95 96
96 97
97 98
98 99
99 100
100 101
101 102
102 103
103 104
104 105
105 106
106 107
107 108
108 109
109 110
110 111
111 112
112 113
113 114
114 115
115 116
116 117
117 118
118 119
119 120
120 121
121 122
122 123
123 124
124 125
125 126
126 127
127 128
128 129
129 130
130 131
131 132
132 133
133 134
134 135
135 136
136 137
137 138
138 139
139 140
140 141
141 142
142 143
143 144
144 145
145 146
146 147
147 148
148 149
149 150
150 151
151 152
152 153
153 154
154 155
155 156
156 157
157 158
158 159
159 160
160 161
161 162
162 163
163 164
164 165
165 166
166 167
167 168
168 169
169 170
170 171
171 172
172 173
173 174
174 175
175 176
176 177
177 178
178 179
179 180
180 181
181 182
182 183
183 184
184 185
185 186
186 187
187 188
188 189
189 190
190 191
191 192
192 193
193 194
194 195
195 196
196 197
197 198
198 199
199 200
200 201
201 202
202 203
203 204
204 205
205 206
206 207
207 208
208 209
209 210
210 211
211 212
212 213
213 214
214 215
215 216
216 217
217 218
218 219
219 220
220 221
221 222
222 223
223 224
224 225
225 226
226 227
227 228
228 229
229 230
230 231
231 232
232 233
233 234
234 235
235 236
236 237
237 238
238 239
239 240
240 241
241 242
242 243
243 244
244 245
245 246
246 247
247 248
248 249
249 250
250 251
251 252
252 253
253 254
254 255
255 256
256 257
257 258
258 259
259 260
260 261
261 262
262 263
263 264
264 265
265 266
266 267
267 268
268 269
269 270
270 271
271 272
272 273
273 274
274 275
275 276
276 277
277 278
278 279
279 280
280 281
281 282
282 283
283 284
284 285
285 286
286 287
287 288
288 289
289 290
290 291
291 292
292 293
293 294
294 295
295 296
296 297
297 298
298 299
299 300
300 301
301 302
302 303
303 304
304 305
305 306
306 307
307 308
308 309
309 310
310 311
311 312
312 313
313 314
314 315
315 316
316 317
317 318
318 319
319 320
320 321
321 322
322 323
323 324
324 325
325 326
326 327
327 328
328 329
329 330
330 331
331 332
332 333
333 334
334 335
335 336
336 337
337 338
338 339
339 340
340 341
341 342
342 343
343 344
344 345
345 346
346 347
347 348
348 349
349 350
350 351
351 352
352 353
353 354
354 355
355 356
356 357
357 358
358 359
359 360
360 361
361 362
362 363
363 364
364 365
365 366
366 367
367 368
368 369
369 370
370 371
371 372
372 373
373 374
374 375
375 376
376 377
377 378
378 379
379 380
380 381
381 382
382 383
383 384
384 385
385 386
386 387
387 388
388 389
389 390
390 391
391 392
392 393
393 394
394 395
395 396
396 397
397 398
398 399
399 400
400 401
401 402
402 403
403 404
404 405
405 406
406 407
407 408
408 409
409 410
410 411
411 412
412 413
413 414
414 415
415 416
416 417
417 418
418 419
419 420
420 421
421 422
422 423
423 424
424 425
425 426
426 427
427 428
428 429
429 430
430 431
431 432
432 433
433 434
434 435
435 436
436 437
437 438
438 439
439 440
440 441
441 442
442 443
443 444
444 445
445 446
446 447
447 448
448 449
449 450
450 451
451 452
452 453
453 454
454 455
455 456
456 457
457 458
458 459
459 460
460 461
461 462
462 463
463 464
464 465
465 466
466 467
467 468
468 469
469 470
470 471
471 472
472 473
473 474
474 475
475 476
476 477
477 478
478 479
479 480
480 481
481 482
482 483
483 484
484 485
485 486
486 487
487 488
488 489
489 490
490 491
491 492
492 493
493 494
494 495
495 496
496 497
497 498
498 499
499 500
500 501
501 502
502 503
503 504
504 505
505 506
506 507
507 508
508 509
509 510
510 511
511 512
512 513
513 514
514 515
515 516
516 517
517 518
518 519
519 520
520 521
521 522
522 523
523 524
524 525
525 526
526 527
527 528
528 529
529 530
530 531
531 532
532 533
533 534
534 535
535 536
536 537
537 538
538 539
539 540
540 541
541 542
542 543
543 544
544 545
545 546
546 547
547 548
548 549
549 550
550 551
551 552
552 553
553 554
554 555
555 556
556 557
557 558
558 559
559 560
560 561
561 562
562 563
563 564
564 565
565 566
566 567
567 568
568 569
569 570
570 571
571 572
572 573
573 574
574 575
575 576
576 577
577 578
578 579
579 580
580 581
581 582
582 583
583 584
584 585
585 586
586 587
587 588
588 589
589 590
590 591
591 592
592 593
593 594
594 595
595 596
596 597
597 598
598 599
599 600
600 601
601 602
602 603
603 604
604 605
605 606
606 607
607 608
608 609
609 610
610 611
611 612
612 613
613 614
614 615
615 616
616 617
617 618
618 619
619 620
620 621
621 622
622 623
623 624
624 625
625 626
626 627
627 628
628 629
629 630
630 631
631 632
632 633
633 634
634 635
635 636
636 637
637 638
638 639
639 640
640 641
641 642
642 643
643 644
644 645
645 646
646 647
647 648
648 649
649 650
650 651
651 652
652 653
653 654
654 655
655 656
656 657
657 658
658 659
659 660
660 661
661 662
662 663
663 664
664 665
665 666
666 667
667 668
668 669
669 670
670 671
671 672
672 673
673 674
674 675
675 676
676 677
677 678
678 679
679 680
680 681
681 682
682 683
683 684
684 685
685 686
686 687
687 688
688 689
689 690
690 691
691 692
692 693
693 694
694 695
695 696
696 697
697 698
698 699
699 700
700 701
701 702
702 703
703 704
704 705
705 706
706 707
707 708
708 709
709 710
710 711
711 712
712 713
713 714
714 715
715 716
716 717
717 718
718 719
719 720
720 721
721 722
722 723
723 724
724 725
725 726
726 727
727 728
728 729
729 730
730 731
731 732
732 733
733 734
734 735
735 736
736 737
737 738
738 739
739 740
740 741
741 742
742 743
743 744
744 745
745 746
746 747
747 748
748 749
749 750
750 751
751 752
752 753
753 754
754 755
755 756
756 757
757 758
758 759
759 760
760 761
761 762
762 763
763 764
764 765
765 766
766 767
767 768
768 769
769 770
770 771
771 772
772 773
773 774
774 775
775 776
776 777
777 778
778 779
779 780
780 781
781 782
782 783
783 784
784 785
785 786
786 787
787 788
788 789
789 790
790 791
791 792
792 793
793 794
794 795
795 796
796 797
797 798
798 799
799 800
800 801
801 802
802 803
803 804
804 805
805 806
806 807
807 808
808 809
809 810
810 811
811 812
812 813
813 814
814 815
815 816
816 817
817 818
818 819
819 820
820 821
821 822
822 823
823 824
824 825
825 826
826 827
827 828
828 829
829 830
830 831
831 832
832 833
833 834
834 835
835 836
836 837
837 838
838 839
839 840
840 841
841 842
842 843
843 844
844 845
845 846
846 847
847 848
848 849
849 850
850 851
851 852
852 853
853 854
854 855
855 856
856 857
857 858
858 859
859 860
860 861
861 862
862 863
863 864
864 865
865 866
866 867
867 868
868 869
869 870
870 871
871 872
872 873
873 874
874 875
875 876
876 877
877 878
878 879
879 880
880 881
881 882
882 883
883 884
884 885
885 886
886 887
887 888
888 889
889 890
890 891
891 892
892 893
893 894
894 895
895 896
896 897
897 898
898 899
899 900
900 901
901 902
902 903
903 904
904 905
905 906
906 907
907 908
908 909
909 910
910 911
911 912
912 913
913 914
914 915
915 916
916 917
917 918
918 919
919 920
920 921
921 922
922 923
923 924
924 925
925 926
926 927
927 928
928 929
929 930
930 931
931 932
932 933
933 934
934 935
935 936
936 937
937 938
938 939
939 940
940 941
941 942
942 943
943 944
944 945
945 946
946 947
947 948
948 949
949 950
950 951
951 952
952 953
953 954
954 955
955 956
956 957
957 958
958 959
959 960
960 961
961 962
962 963
963 964
964 965
965 966
966 967
967 968
968 969
969 970
970 971
971 972
972 973
973 974
974 975
975 976
976 977
977 978
978 979
979 980
980 981
981 982
982 983
983 984
984 985
985 986
986 987
987 988
988 989
989 990
990 991
991 992
992 993
993 994
994 995
995 996
996 997
997 998
998 999
999 1000
1000 1001
1001 1002
1002 1003
1003 1004
1004 1005
1005 1006
1006 1007
1007 1008
1008 1009
1009 1010
1010 1011
1011 1012
1012 1013
1013 1014
1014 1015
1015 1016
1016 1017
1017 1018
1018 1019
1019 1020
1020 1021
1021 1022
1022 1023
1023 1024
1024 1025
1025 1026
1026 1027
1027 1028
1028 1029
1029 1030
1030 1031
1031 1032
1032 1033
1033 1034
1034 1035
1035 1036
1036 1037
1037 1038
1038 1039
1039 1040
1040 1041
1041 1042
1042 1043
1043 1044
1044 1045
1045 1046
1046 1047
1047 1048
1048 1049
1049 1050
1050 1051
1051 1052
1052 1053
1053 1054
1054 1055
1055 1056
1056 1057
1057 1058
1058 1059
1059 1060
1060 1061
1061 1062
1062 1063
1063 1064
1064 1065
1065 1066
1066 1067
1067 1068
1068 1069
1069 1070
1070 1071
1071 1072
1072 1073
1073 1074
1074 1075
1075 1076
1076 1077
1077 1078
1078 1079
1079 1080
1080 1081
1081 1082
1082 1083
1083 1084
1084 1085
1085 1086
1086 1087
1087 1088
1088 1089
1089 1090
1090 1091
1091 1092
1092 1093
1093 1094
1094 1095
1095 1096
1096 1097
1097 1098
1098 1099
1099 1100
1100 1101
1101 1102
1102 1103
1103 1104
1104 1105
1105 1106
1106 1107
1107 1108
1108 1109
1109 1110
1110 1111
1111 1112
1112 1113
1113 1114
1114 1115
1115 1116
1116 1117
1117 1118
1118 1119
1119 1120
1120 1121
1121 1122
1122 1123
1123 1124
1124 1125
1125 1126
1126 1127
1127 1128
1128 1129
1129 1130
1130 1131
1131 1132
1132 1133
1133 1134
1134 1135
1135 1136
1136 1137
1137 1138
1138 1139
1139 1140
1140 1141
1141 1142
1142 1143
1143 1144
1144 1145
1145 1146
1146 1147
1147 1148
1148 1149
1149 1150
1150 1151
1151 1152
1152 1153
1153 1154
1154 1155
1155 1156
1156 1157
1157 1158
1158 1159
1159 1160
1160 1161
1161 1162
1162 1163
1163 1164
1164 1165
1165 1166
1166 1167
1167 1168
1168 1169
1169 1170
1170 1171
1171 1172
1172 1173
1173 1174
1174 1175
1175 1176
1176 1177
1177 1178
1178 1179
1179 1180
1180 1181
1181 1182
1182 1183
1183 1184
1184 1185
1185 1186
1186 1187
1187 1188
1188 1189
1189 1190
1190 1191
1191 1192
1192 1193
1193 1194
1194 1195
1195 1196
1196 1197
1197 1198
1198 1199
1199 1200
1200 1201
1201 1202
1202 1203
1203 1204
1204 1205
1205 1206
1206 1207
1207 1208
1208 1209
1209 1210
1210 1211
1211 1212
1212 1213
1213 1214
1214 1215
1215 1216
1216 1217
1217 1218
1218 1219
1219 1220
1220 1221
1221 1222
1222 1223
1223 1224
1224 1225
1225 1226
1226 1227
1227 1228
1228 1229
1229 1230
1230 1231
1231 1232
1232 1233
1233 1234
1234 1235
1235 1236
1236 1237
1237 1238
1238 1239
1239 1240
1240 1241
1241 1242
1242 1243
1243 1244
1244 1245
1245 1246
1246 1247
1247 1248
1248 1249
1249 1250
1250 1251
1251 1252
1252 1253
1253 1254
1254 1255
1255 1256
1256 1257
1257 1258
1258 1259
1259 1260
1260 1261
1261 1262
1262 1263
1263 1264
1264 1265
1265 1266
1266 1267
1267 1268
1268 1269
1269 1270
1270 1271
1271 1272
1272 1273
1273 1274
1274 1275
1275 1276
1276 1277
1277 1278
1278 1279
1279 1280
1280 1281
1281 1282
1282 1283
1283 1284
1284 1285
1285 1286
1286 1287
1287 1288
1288 1289
1289 1290
1290 1291
1291 1292
1292 1293
1293 1294
1294 1295
1295 1296
1296 1297
1297 1298
1298 1299
1299 1300
1300 1301
1301 1302
1302 1303
1303 1304
1304 1305
1305 1306
1306 1307
1307 1308
1308 1309
1309 1310
1310 1311
1311 1312
1312 1313
1313 1314
1314 1315
1315 1316
1316 1317
1317 1318
1318 1319
1319 1320
1320 1321
1321 1322
1322 1323
1323 1324
1324 1325
1325 1326
1326 1327
1327 1328
1328 1329
1329 1330
1330 1331
1331 1332
1332 1333
1333 1334
1334 1335
1335 1336
1336 1337
1337 1338
1338 1339
1339 1340
1340 1341
1341 1342
1342 1343
1343 1344
1344 1345
1345 1346
1346 1347
1347 1348
1348 1349
1349 1350
1350 1351
1351 1352
1352 1353
1353 1354
1354 1355
1355 1356
1356 1357
1357 1358
1358 1359
1359 1360
1360 1361
1361 1362
1362 1363
1363 1364
1364 1365
1365 1366
1366 1367
1367 1368
1368 1369
1369 1370
1370 1371
1371 1372
1372 1373
1373 1374
1374 1375
1375 1376
1376 1377
1377 1378
1378 1379
1379 1380
1380 1381
1381 1382
1382 1383
1383 1384
1384 1385
1385 1386
1386 1387
13
```



**Password is**

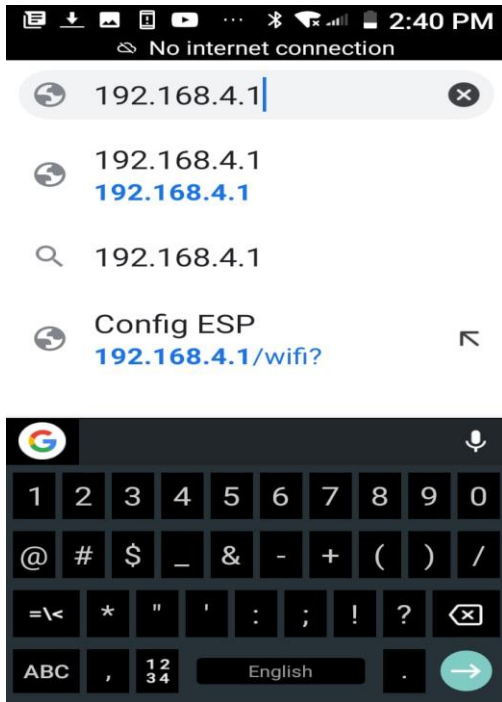
**12345678**



**5.After connecting the Robato Hotspot in the same mobile open**

**Any web browser like chrome type this local ip in the search bar**

**192.168.4.1**



**Configure the everything in the webpage,**

**Type the ip address you got from the mobile Bluetooth in the gateway address  
column,**

**Username**

**Password**

**These columns should be filled accordingly by which you have configured in  
local mqtt username and password**



⚠️ 192.168.4.1



Hello from Robato Systems

scan

192.168.4.1

1. stijo (-46)\*
2. DIRECT-OCDESKTOP-T9RV88GmsFQ (-38)\*

SSID: stijo ▼

Password:

stijoseph

Choose Room:

Lobby ▼

Choose Device:

Device3 ▼

Location:

Location

Gateway Address:

192.168.43.81

Username:

username

User password:

password

Choose Appliance1:

Light1 ▼

Choose Appliance2:

Fan1 ▼

Choose Appliance3:

Fan8 ▼

Choose Appliance4:

Light7 ▼

Submit

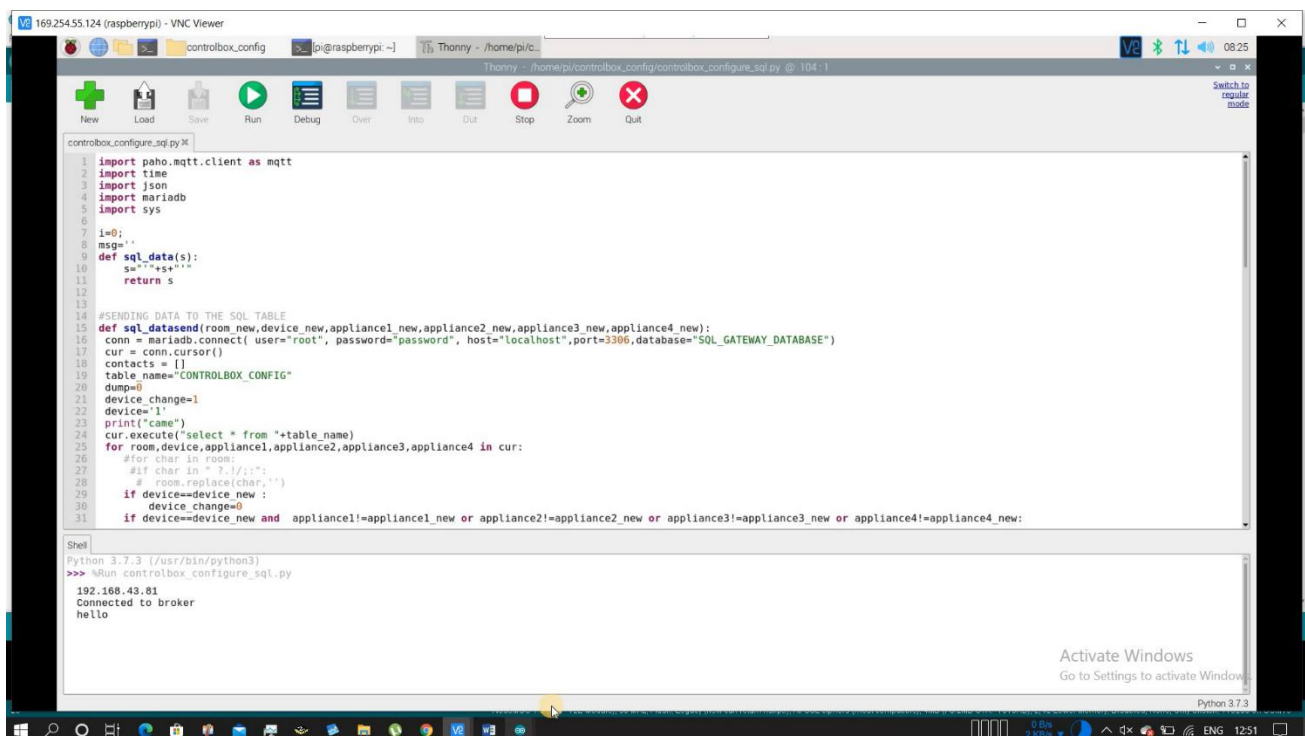
```
device_with_customdesign | Arduino 1.8.10 Hourly Build 2019/05/21 09:33
File Edit Sketch Tools Help
COM16
device_wit
1 /* 14:34:24.216 ->
2 If yo 14:34:25.203 -> Starting up the device
3 If yo 14:34:38.624 -> Turning the HotSpot On
4 In ho 14:34:38.624 -> Local IP: (IP unset)
5 To co 14:34:38.624 -> SoftAP IP: 192.168.4.1
6 {"Use 14:34:38.624 -> Server started
7 For e 14:34:40.992 -> scan done
8 Devic 14:34:40.992 -> 2 networks found
9 To tu 14:34:40.992 -> 1: stijo (-46)*
10 To re 14:34:41.026 -> 2: DIRECT-OCDESKTOP-T9RV88GmsFQ (-38)*
11 /* 14:34:41.026 ->
12 14:34:41.195 -> softap
13 #incl 14:34:41.195 -> Local IP: (IP unset)
14 #incl 14:34:41.195 -> SoftAP IP: 192.168.4.1
15 #incl 14:34:41.195 -> Server started
16 #incl 14:34:41.195 -> over
17 #incl 14:34:41.195 ->
18 #incl 14:34:41.195 -> Waiting.
19 14:34:41.195 -> .....
20 //var 14:41:24.177 -> Password is: stijojooseph
21 int i 14:41:24.177 -> Romm name is: Lobby
22 int s 14:41:24.177 -> Device name is: Device3
23 const 14:41:24.177 -> Location is: Location
24 const 14:41:24.177 -> Username is: username
25 Strin 14:41:24.177 -> Userpass is: password
26 Strin 14:41:24.177 -> Appliance1 is: Light1
27 14:41:24.177 -> Appliance2 is: Fan1
28 // Ga 14:41:24.177 -> Appliance3 is: Fan8
29 char 14:41:24.177 -> Appliance4 is: Light7
30 char 14:41:24.177 -> Clearing FS
31 char
32 char
33 char
34 char
Leaving..
Hard rese
Autoscroll Show timestamp
Newline 115200 baud
```

**If you Receive the msg “config.file “ is writen successfully then connect D0 pin to GND in a fastway less than 6 seconds to stop the hotspot and connect to the configured wifi**

## 4 Gateway configurations to Database

. Gateway will store the control box information in a table name `control_box_configuration`.

- **FOR CHECKING THAT**
  - OPEN THE FOLDER CONTROLBOX\_CONFIG**
  - **RUN THE FILE IN THONNY IDE**
- `controlbox_configure_sql.py`**

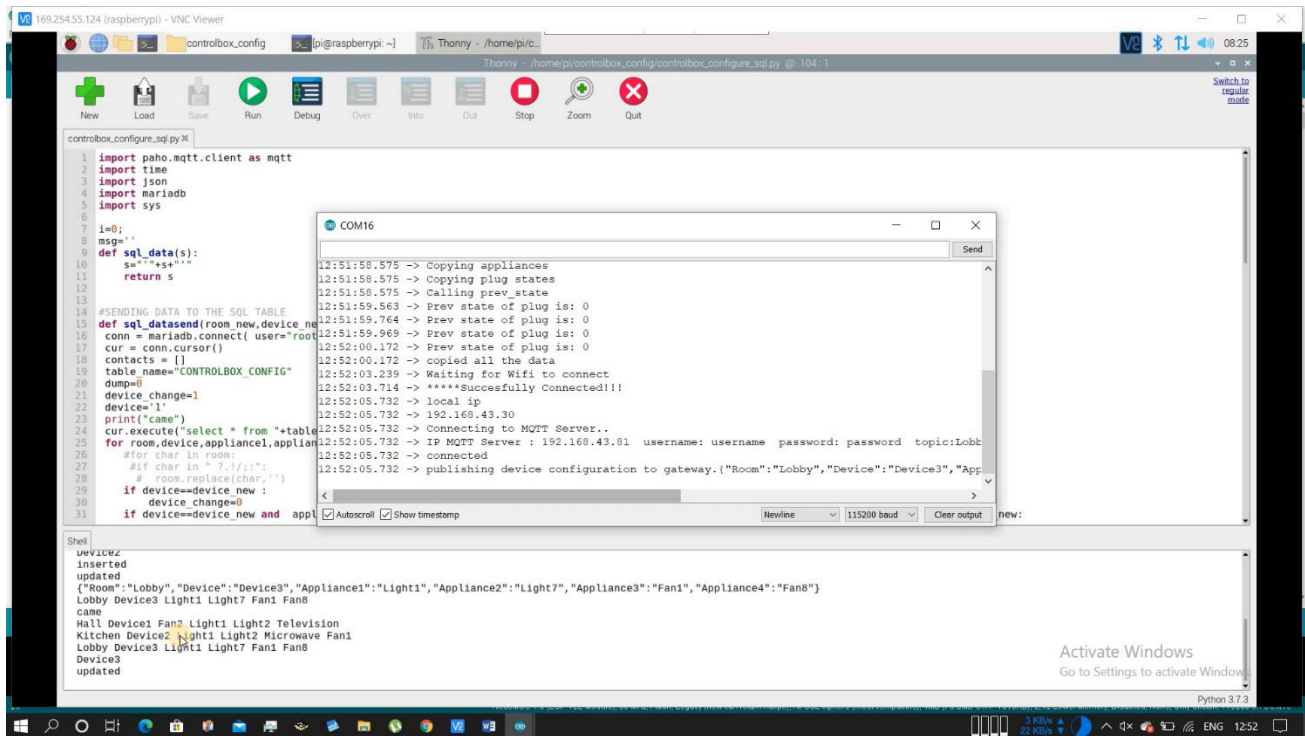


```
controlbox_configure_sql.py
1 import paho.mqtt.client as mqtt
2 import time
3 import json
4 import mariadb
5 import sys
6
7 i=0;
8 msg=""
9
10 def sql_data(s):
11     sw=""
12     return s
13
14 #SENDING DATA TO THE SQL TABLE
15 def sql_datasend(room_new,device_new,appliance1_new,appliance2_new,appliance3_new,appliance4_new):
16     conn = mariadb.connect( user="root", password="password", host="localhost",port=3306,database="SQL_GATEWAY_DATABASE")
17     cur = conn.cursor()
18     contacts = {}
19     table_name="CONTROLBOX_CONFIG"
20     dump=0
21     device_change=1
22     device="1"
23     print("came")
24     cur.execute("select * from "+table_name)
25     for room,device,appliance1,appliance2,appliance3,appliance4 in cur:
26         #for char in room:
27             #if char in "7.//:;":
28                 # room.replace(char,"")
29             if device==device_new:
30                 device_change=0
31             if device==device_new and appliance1!=appliance1_new or appliance2!=appliance2_new or appliance3!=appliance3_new or appliance4!=appliance4_new:
```

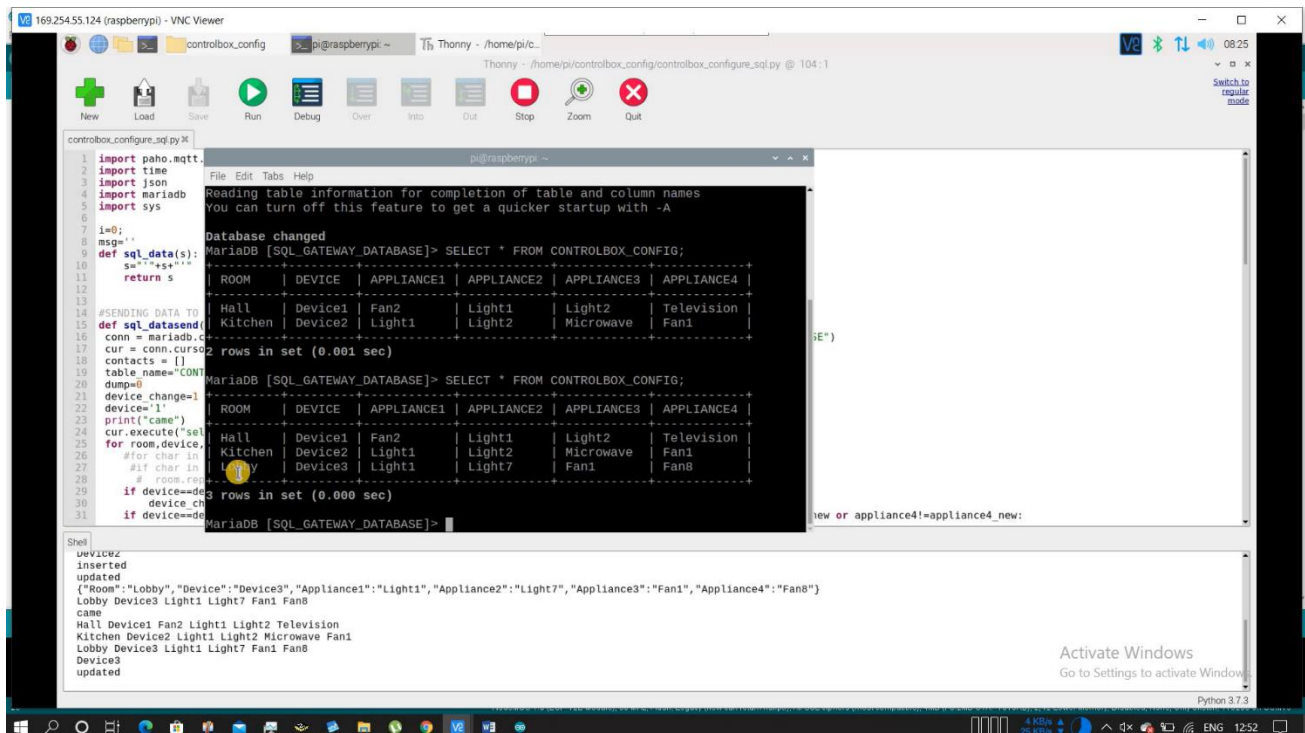
```
Shell
Python 3.7.3 (/usr/bin/python3)
>>> %Run controlbox_configure_sql.py
192.168.43.81
Connected to broker
hello
```

(remember D0 pin should be connected to GND only )

next connect your esp8266 to pc and press reset



Verify with the database





## 5. Mobile - Gateway – ControlBox

### Control appliance

Step 1- To ON/OFF appliance User will send a JSON message with topic Room/ControlBox

JSON format will be-

```
{ "Username": "abcd", "AccessKey": "xyz", "Location": "xyz"  
  "ApplianceName": "Appliance", "state": "ON/OFF" }
```

Or

```
{ "Username": "abcd", "AccessKey": "xyz", "Location": "xyz", "ControlBox":  
  "BoxName", "ApplianceName": "appliance", "speed": "0/1/2..." }
```

**\*\* Here if speed is 0, means appliance is in OFF condition**

Step 2- The appliance will be ON or OFF based on message also device will send an acknowledgement message to user that their command has been executed successfully.

Acknowledgement message will be published with topic Acknowledgement/Room/ControlBox.

JSON message will be- { "Username": "abcd", "AccessKey": "xyz",  
 "Location": "xyz", "ApplianceName": "Appliance",  
 "state": "ON/OFF" }

Or

```
{ "Username": "abcd", "AccessKey": "xyz", "Location": "xyz", "ControlBox":  
  "BoxName", "ApplianceName": "appliance", "speed": "0/1/2..." }
```

Step 3- When device will send the acknowledgement message, gateway will save the current device state in table named Room\_ControlBox\_Appliance. And the columns in the table will be-

Date	Location	Client	Appliance	ON time	OFF time	Total ON time

**Update security key (admin user only)**

**Step 1- Only admin can update the security key. To update the security key admin need to open their app and go into gateway setting.**

**Step 2- There will be an option to update security key, when user click on the option, gateway will response with a webpage secured with updated user id and password.**

**Step 3- User need to login into the webpage with its user id and password, if it is correct webpage will provide option to update security key.**

**Step 4- User will enter new security key and save it.**

**Step 5- Gateway will update the security key in MySQL table user\_data.**

**Step 6- Since gateway is in sync with cloud, security will also be updated in cloud. Also, if you change security key in cloud it will updated in gateway.**

#### **VIDEO LINK-**

**<https://drive.google.com/file/d/1arGjy2AjAy94Mkp11ctuQJ2PHv87-6wA/view?usp=sharing>**

#### **Json format**

```
{"Username":"your_username", "AccessKey":"xyz",  
"Location":"Room_name", "Appliance":"Appliance_name", "state":"0/1"}
```

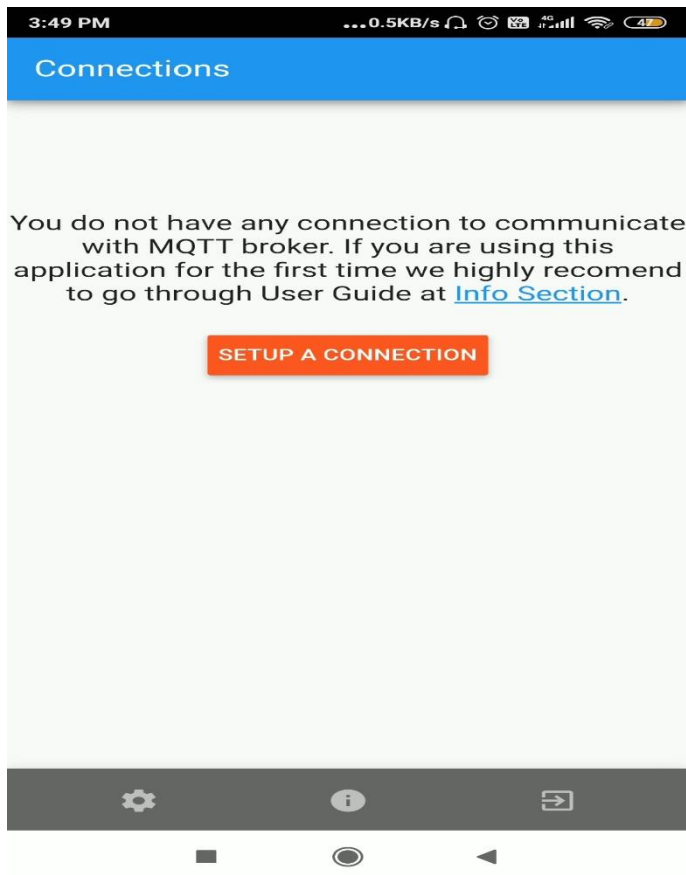
#### **Example**

```
{"Username":"username1", "AccessKey":"xyz", "Location":"Lobby",  
"Appliance":"Light1", "state":"5"}
```

#### **Download the app**

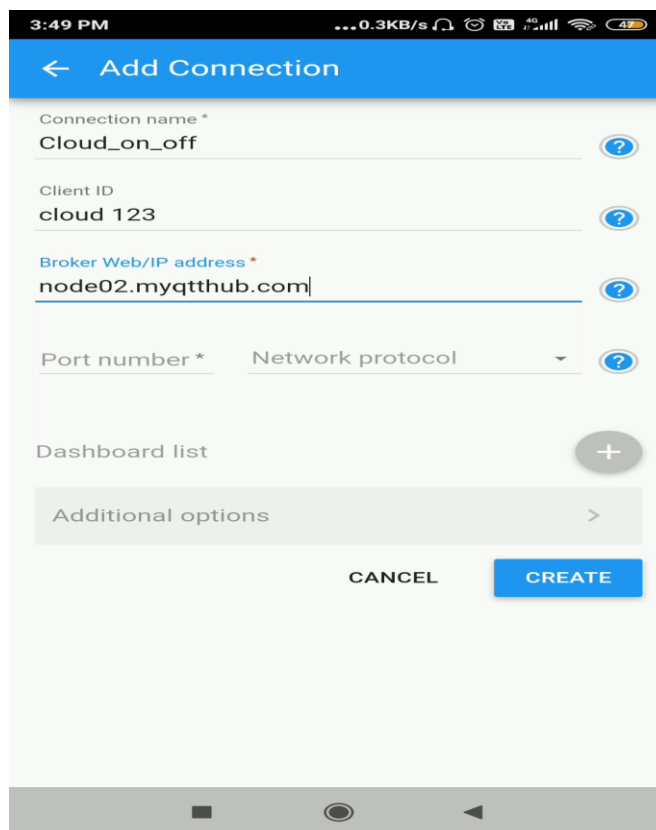
**<https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod>**

**open the app**



Lets configure cloud details

first



Type your  
bootstrap  
credentials  
username and  
password here

3:49 PM ...0.1KB/s

← Add Connection

Port number \* 1883 Network protocol TCP

Dashboard list +

Additional options v

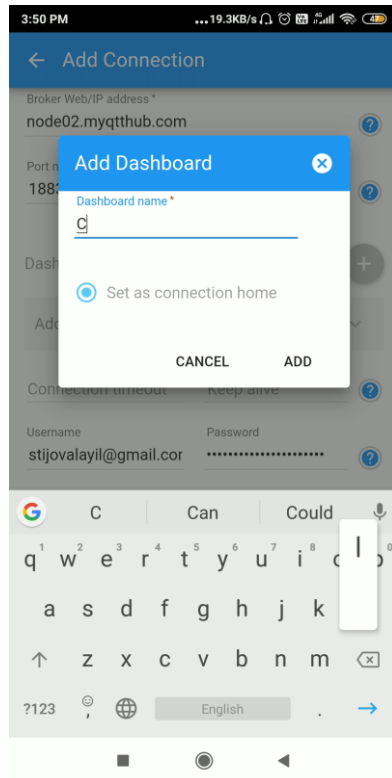
Connection timeout Keep alive

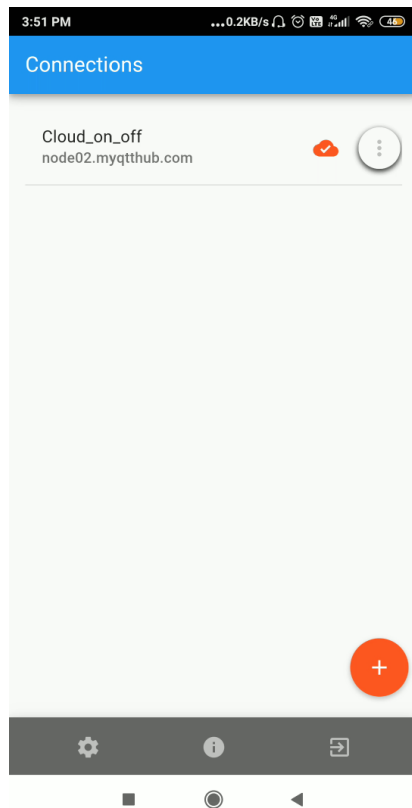
Username stijovalayil@gmail.com Password

Add will message +

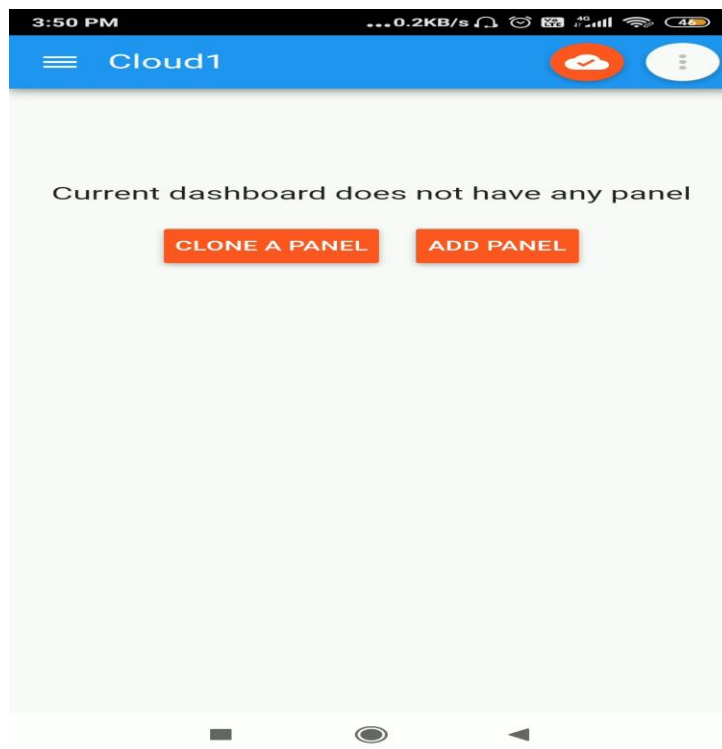
1 2 3 4 5 6 7 8 9 0  
q w e r t y u i o p  
a s d f g h j k l  
↑ z x c v b n m  
?123 , English .

**Add dashboard any name**

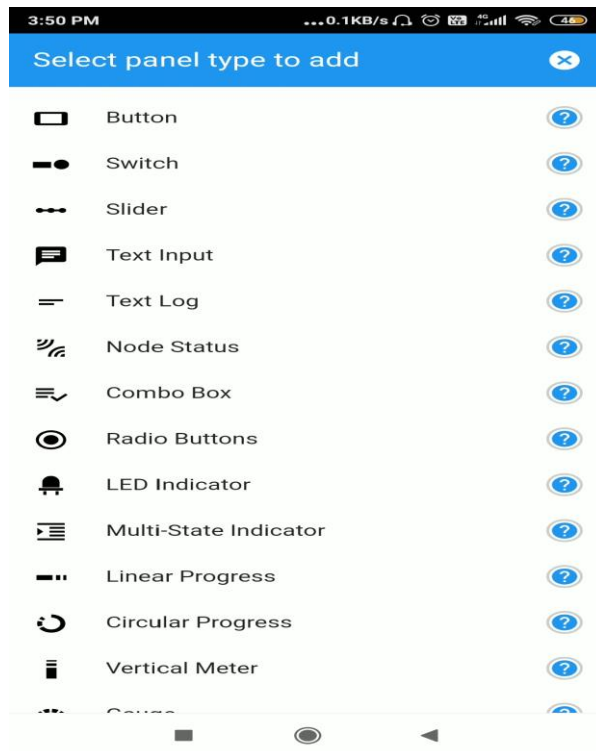




## Add panel



## Press on switch



## Topic

### Lobby/Device3

#### Payload on

```
{"Username":"username1",  
  "AccessKey":"xyz",  
  "Location":"Lobby",  
  "Appliance":"Light1", "state":"5"}
```

#### Payload off

```
{"Username": "username1",  
  "AccessKey": "xyz",  
  "Location": "Lobby",  
  "Appliance": "Light1",  
  "state": "0"}
```

3:53 PM ...2.6KB/s

← Add a Switch panel

Panel name \*  
Light1

Topic \*  
Lobby/Device3

Subscribe Topic ?

Payload on \*  
{ "Username": "username1", "AccessKey": "xyz",  
Payload off \*  
"Location": "Lobby", "Appliance": "Light1", "state": "5" }

☐ Use icon switch

and the to

q<sup>1</sup> w<sup>2</sup> e<sup>3</sup> r<sup>4</sup> t<sup>5</sup> y<sup>6</sup> u<sup>7</sup> i<sup>8</sup> o<sup>9</sup> p<sup>0</sup>

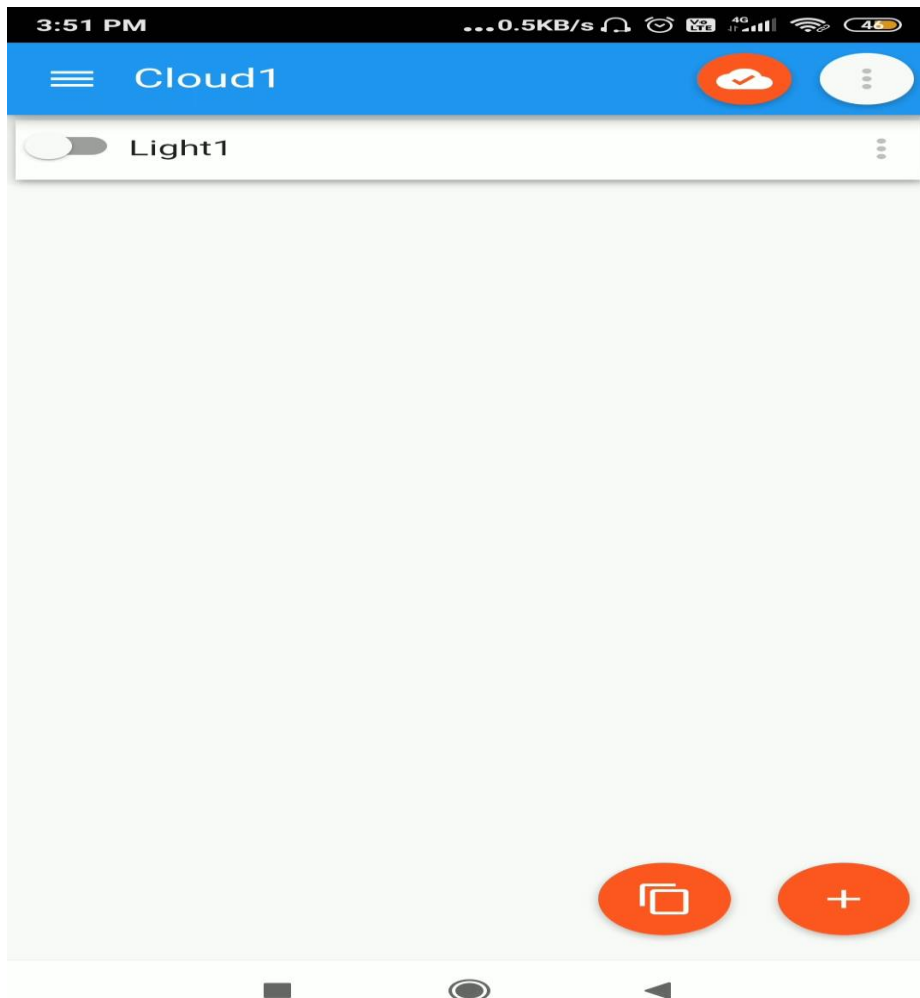
a s d f g h j k l

↑ z x c v b n m < x

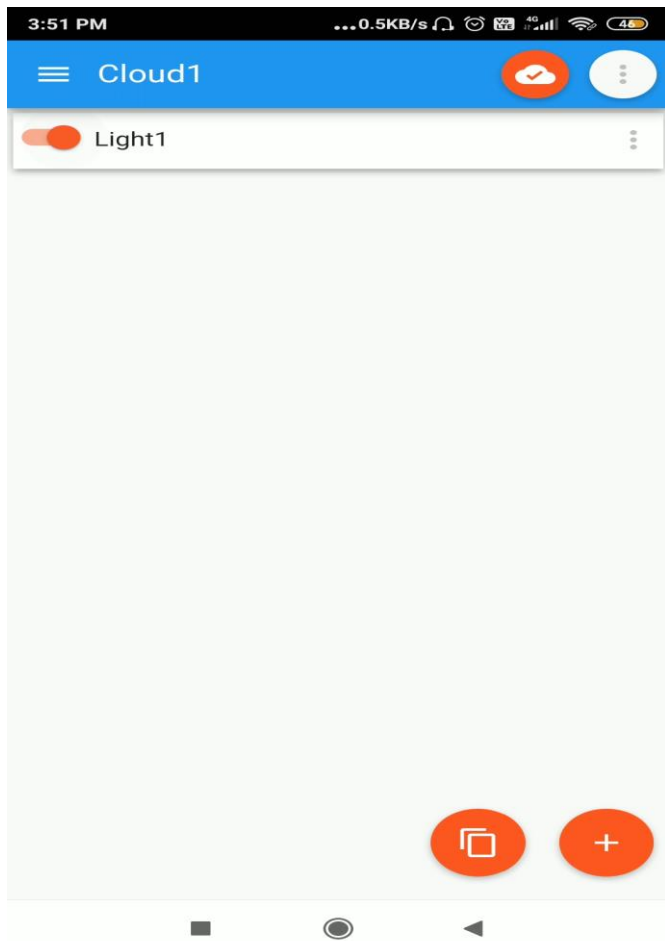
?123 , . English →

save





**Turn on light slide it**



**Light will turn on**

**Similarly configure for**

**local,**

**Port 1884**

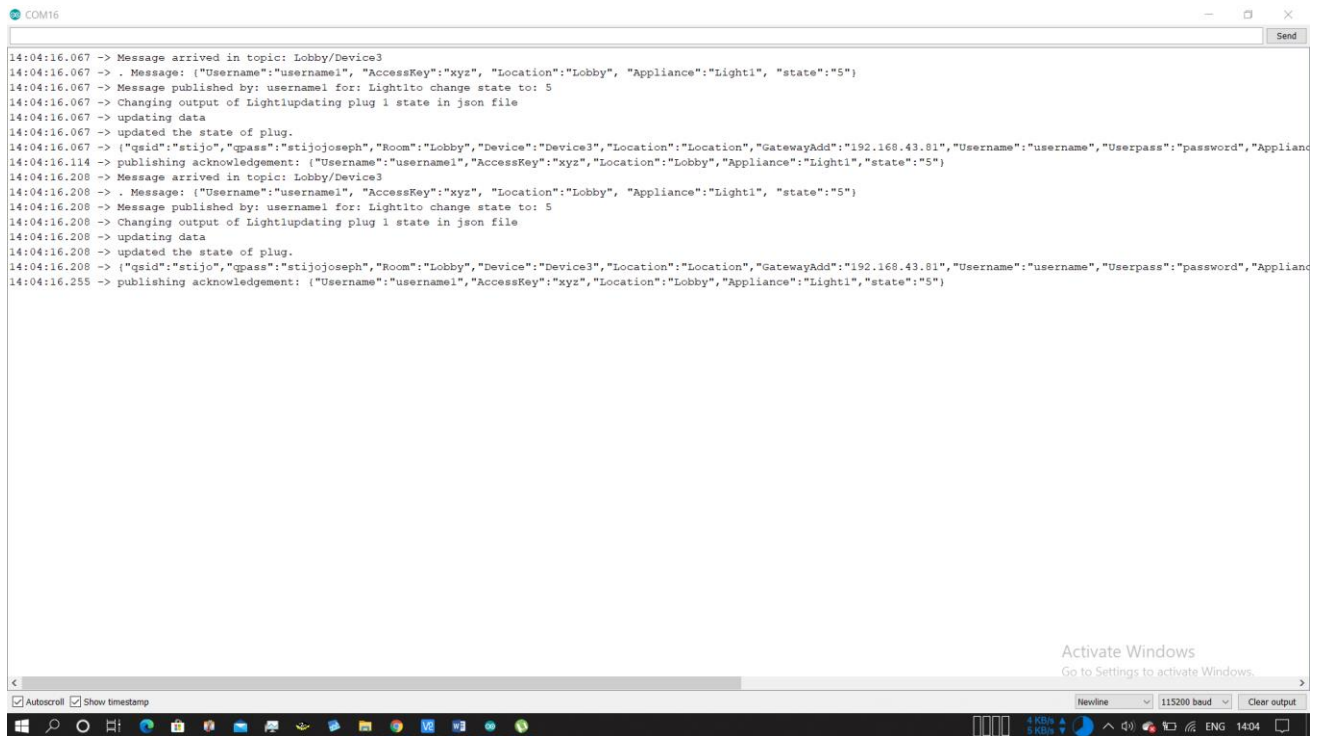
**Username**

**password will be local**

**mosquito broker username**

**and password**

## check the serial monitor for light1 turning on status



The screenshot shows a serial monitor window titled 'COM16'. The window displays a series of MQTT messages and system logs. The messages are as follows:

```
14:04:16.067 -> Message arrived in topic: Lobby/Device3
14:04:16.067 -> . Message: {"Username":"username1", "AccessKey":"xyz", "Location":"Lobby", "Appliance":"Light1", "state":"5"}
14:04:16.067 -> Message published by: username1 for: Light1to change state to: 5
14:04:16.067 -> Changing output of Light1updating plug 1 state in json file
14:04:16.067 -> updating data
14:04:16.067 -> updated the state of plug.
14:04:16.067 -> {"qeid":"stijo","qpass":"stiojoseph","Room":"Lobby","Device":"Device3","Location":"Location","GatewayAdd":"192.168.43.81","Username":"username","Userpass":"password","Appliance":"Light1","state":"5"}
14:04:16.114 -> publishing acknowledgement: {"Username":"username1","AccessKey":"xyz","Location":"Lobby","Appliance":"Light1","state":"5"}
14:04:16.208 -> Message arrived in topic: Lobby/Device3
14:04:16.208 -> . Message: {"Username":"username1", "AccessKey":"xyz", "Location":"Lobby", "Appliance":"Light1", "state":"5"}
14:04:16.208 -> Message published by: username1 for: Light1to change state to: 5
14:04:16.208 -> Changing output of Light1updating plug 1 state in json file
14:04:16.208 -> updating data
14:04:16.208 -> updated the state of plug.
14:04:16.208 -> {"qeid":"stijo","qpass":"stiojoseph","Room":"Lobby","Device":"Device3","Location":"Location","GatewayAdd":"192.168.43.81","Username":"username","Userpass":"password","Appliance":"Light1","state":"5"}
14:04:16.255 -> publishing acknowledgement: {"Username":"username1","AccessKey":"xyz","Location":"Lobby","Appliance":"Light1","state":"5"}
```

At the bottom of the window, there is a status bar with the following information:

- Autoscroll: ☒ Show timestamp: ☒
- Serial speed: 115200 baud
- Clear output button

The Windows taskbar is visible at the bottom of the screen, showing the Start button, search icon, and several open applications. The system tray on the right shows the date and time as 14:04.

### 3. Mobile – Cloud

#### Account creation

##### Step 1- T

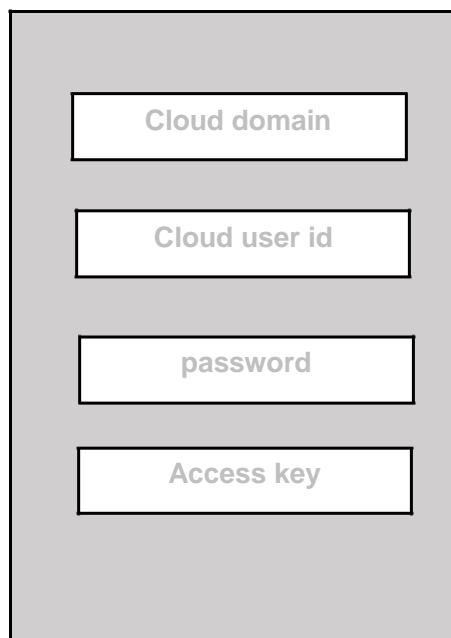
o access the gateway and control the devices user need to create an account in the cloud using app through a webpage.

**Step 2-** To create an account user need to provide some basic information like Name, Mobile no., e-mail, Location etc. and it will be verified by OTP.

**Step 3-** Now user will get a domain name, user id and password, access key from the cloud server.

**Step 4-** Again user come to the local mqtt broker (gateway) and choose option to connect with cloud in app.

**Step 5-** Gateway will ask for domain name, user id and password which was provided by cloud.



Cloud domain

Cloud user id

password

Access key

Save

**Step 6- User will enter the above credential and click on save option.**

**Step 7- Using HTTP, Gateway will send a request to domain name with user id and password to connect with the cloud.**

**Step 8- If credentials provided by gateway is correct, it will connect with the cloud.**

**Step 9- Now cloud will ask gateway for saved data like, user\_data, control box configuration, usage history etc. through a web page. Gateway will sync the data with the cloud.**

### **3.Get appliance usage history and analytics** **For local and cloud**

**Step 1- To get the history of a appliance attached with a control box, admin User app will publish a message to gateway/cloud with Topic History/Room/ControlBox .**

**JSON message will be-**

```
{ "Username": "abcd", "AccessKey": "xyz", "Location": "xyz",  
  "ControlBox": "Boxname", "state": "History", "StartDate": "date", "EndDate": "date" }
```

**Example:**

```
{"Username": "username1", "AccessKey": "xyz", "Location":  
"Lobby", "Appliance": "t1", "from_date": "30/04/2021", "to_date": "29/05/2021"}
```

**Step 2- Since all the information present in the gateway is always in sync with cloud, cloud will response the user request.**

**Step 3- Cloud/gateway will retrieve the information from the MySQL database table Room\_ControlBox\_History.**

**Step 4- Cloud will send appliance usage history to the user using a **webpage** and app will show it accordingly**

**MODIFICATION: NOT WEBPAGE , HISTORY OF APPLIANCE IS PUBLISHED BACK IN JSON FORMAT THROUGH MQTT ITSELF**

**Msg received back, format**

```
{ "Username": "user_name" , "AccessKey": "access_key" , "history": [ { "DATE": "dateformat" ,  
"LOCATION": "room_name" , "APPLIANCE": "appliance_name" , "USERNAME":  
"username1" , "ON_TIME": " time_format " , "OFF_TIME": "time_format" ,  
"TOTAL_ON_TIME": " time_format " } , { "DATE": " dateformat " , "LOCATION": "  
room_name " , "APPLIANCE": " appliance_name " , "USERNAME": "username1" ,  
"ON_TIME": " time_format " , "OFF_TIME": " time_format " , "TOTAL_ON_TIME": "  
time_format " } } }
```

```
{ "Username": "username1" , "AccessKey": "xyz" , "history": [ { "DATE": "05/05/2021" ,  
"LOCATION": "Lobby" , "APPLIANCE": "Light7" , "USERNAME": "username1" ,  
"ON_TIME": "08:04:37" , "OFF_TIME": "NULL" , "TOTAL_ON_TIME": "NULL" } , { "DATE":  
"05/05/2021" , "LOCATION": "Lobby" , "APPLIANCE": "Light7" , "USERNAME":  
"username1" , "ON_TIME": "07:44:17" , "OFF_TIME": "07:46:04" , "TOTAL_ON_TIME":  
"0HRS2MINS" } } }
```

**In here time\_format is NULL in some places which means the appliance is not switched off yet**

**If appliance is not turned on or off anytime after the controlbox configurations and**

**The history request is given to database**

**The response will be**

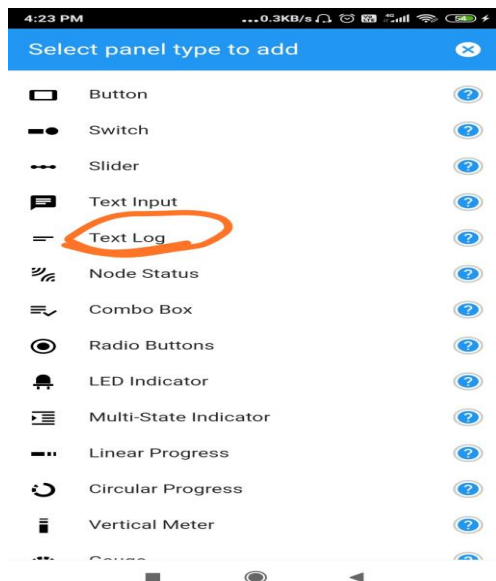
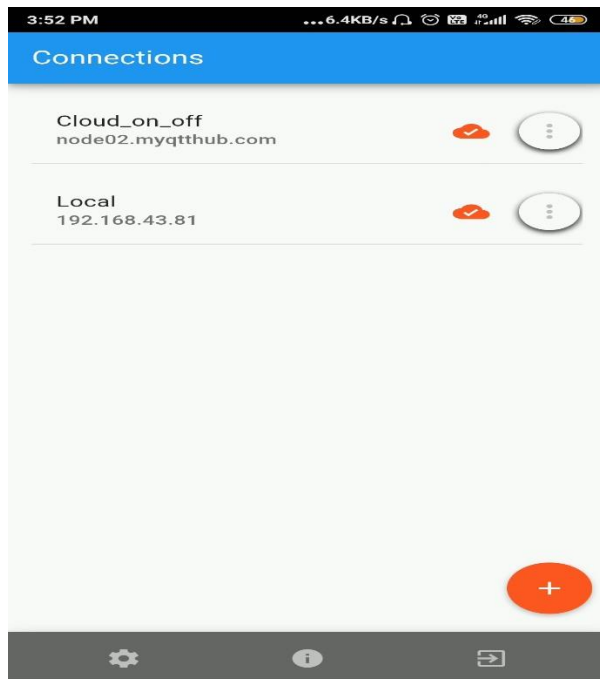
```
{ "Username": "username1" , "AccessKey": "xyz" , "history": [ { "LOCATION": "Lobby" , "APPLIANCE":  
"Fan2" , "STATUS": "NO_HISTORY" } } }
```

**Procedure setup on app**

Select any cloud or local

Video link -

<https://drive.google.com/file/d/1ppRIsmISWe8mO075NI9Pft9IptD5OKIF/view?usp=sharing>



4:25 PM ... 18.8KB/s

← Add a Text Log panel

Panel name \*  
History

Topic \*  
history/pub

Additional options >

QoS  
2

☐ Enable notification ?

☐ Payload is JSON Data

CANCEL CREATE

Next for send the request to history of particular appliance setup

4:25 PM ... 0.3KB/s

Select panel type to add

- Slider ?
- Text Input ?
- Text Log ?
- Node Status ?
- Combo Box ?
- Radio Buttons ?
- LED Indicator ?
- Multi-State Indicator ?
- Linear Progress ?
- Circular Progress ?
- Vertical Meter ?
- Gauge ?
- Color Picker ?

Image add in fututre



## Final result



## 4.Mobile - Cloud – Gateway Get Live Appliances Status

### Get Live Appliances Status

Step 1- To get the live appliance status, user will publish a JSON format message to the control box with topic Room/ControlBox

JSON message will be –

```
{ "Username": "abcd", "AccessKey": "xyz", "Location": "xyz",  
  "ApplianceName": "Appliance", "state": "status" }
```

### Example

```
{ "Username": "username1", "AccessKey": "xyz", "location": "Lobby", "appliance":  
  "Light1" }
```

Step 2- Now control box will reply with appliance status with topic Status/Room/ControlBox.

JSON message will be-

```
{ "Username": "abcd", "AccessKey": "xyz", "Location": "xyz",  
  "ApplianceName": "Appliance", "state": "ON/OFF" }
```

Or

```
{ "Username": "abcd", "AccessKey": "xyz", "Location": "xyz", "ControlBox":  
  "BoxName", "ApplianceName": "appliance", "speed": "0/1/2..." }
```

Step 3- App will show this received data.

### FOR INDIVIDUAL REQUEST THE RESPONSE WILL BE

```
{ "Username": "username1", "AccessKey": "xyz", "status": {  
  "location": "Lobby", "appliance": "Light1", "status": "OFF" } }
```

Or

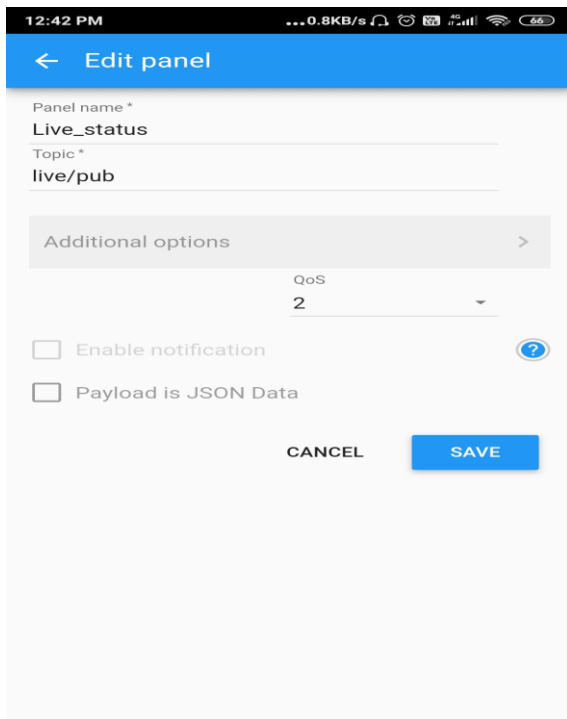
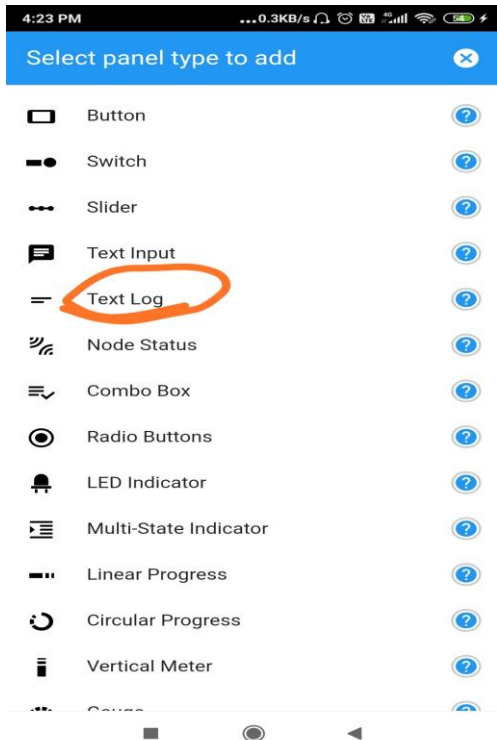
If appliance is equal to ALL

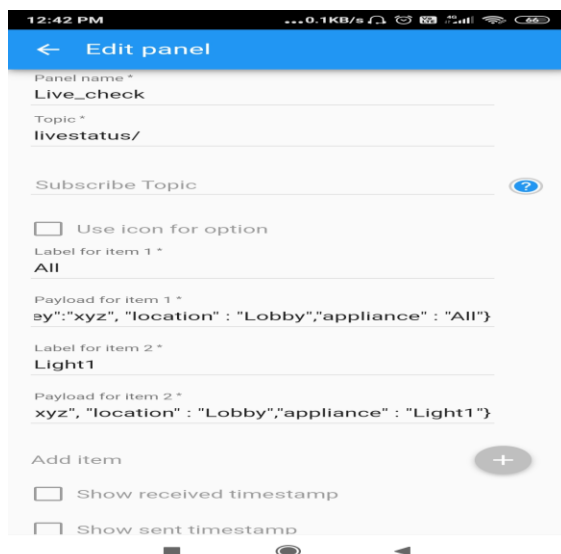
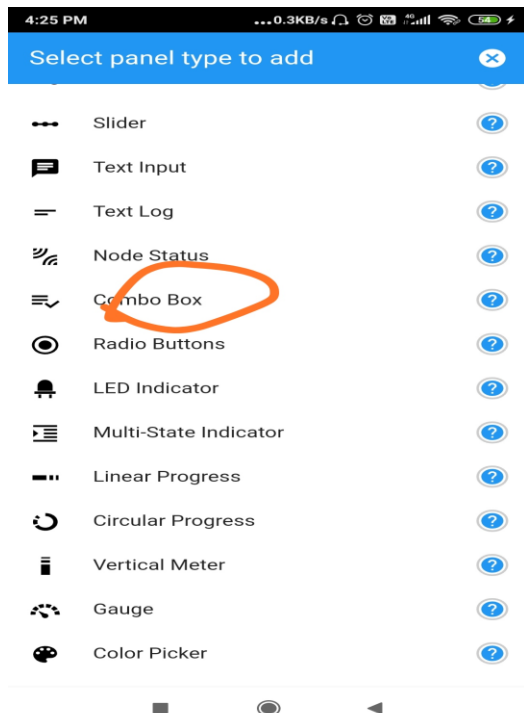
```
{ "Username": "username1", "AccessKey": "xyz", "status": { "Device1": [{  
  "location": "Hall", "appliance": "Fan2", "status": "OFF"}, {  
  "location": "Hall", "appliance": "Light1", "status": "OFF"}, {  
  "location": "Hall", "appliance": "Light2", "status": "OFF"}, {  
  "location": "Hall", "appliance": "Television", "status": "OFF"}]}, { "Device2": [{  
  "location": "Kitchen", "appliance": "Light1", "status": "OFF"}, {  
  "location": "Kitchen", "appliance": "Light2", "status": "OFF"}, {  
  "location": "Kitchen", "appliance": "Microwave", "status": "OFF"}, {  
  "location": "Kitchen", "appliance": "Fan1", "status": "OFF"}]}, { "Device3": [{  
  "location": "Lobby", "appliance": "Light1", "status": "OFF"}, {  
  "location": "Lobby", "appliance": "Light7", "status": "ON"}, {  
  "location": "Lobby", "appliance": "Fan1", "status": "OFF"}, {  
  "location": "Lobby", "appliance": "Fan8", "status": "OFF"}]}}
```

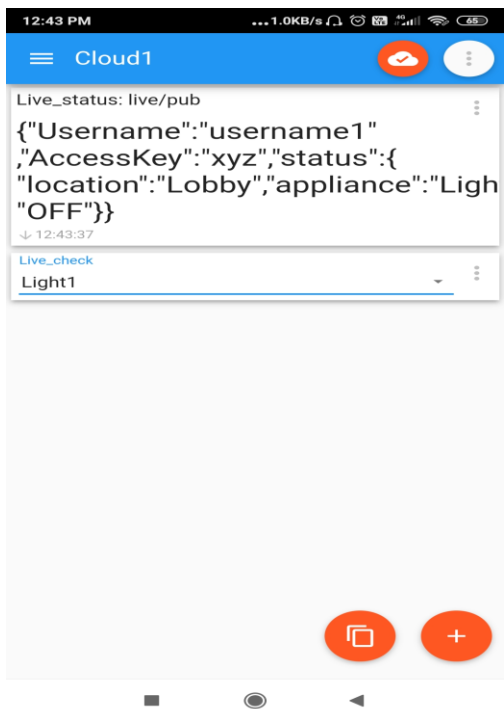
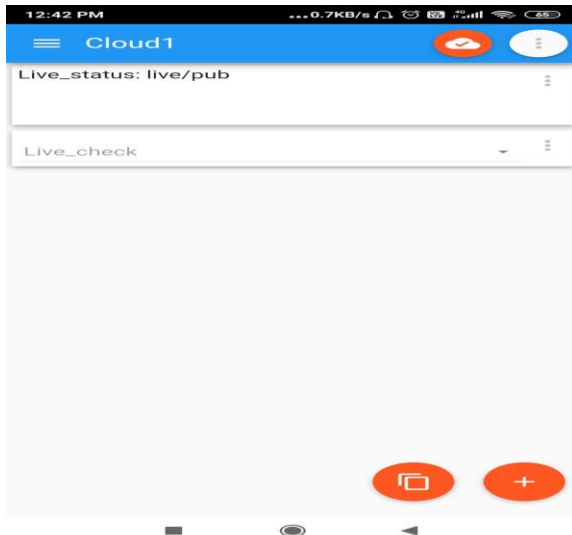
**Video link-**

<https://drive.google.com/file/d/1NtJzZTcrvPJpSnzoXvJLAwcv89MU4Yp2/view?usp=sharing>

**PROCEDURE TO SET UP PANEL**







**OPEN LIVESTATUS\_CHECK FOLDER  
EXECUTE THE livestatus.py file**

**NOTE:**

**AFTER VERIFYING ALL THESE PYTHON FILES AND CONFIGURATIONS**

**YOU CAN EXECUTE A SINGLE PYTHON FILE mainprogram.py**

**Which handles all the tasks which you have individually verified now**

**mainprogram.py**

**and sent the request from the panels in the app**

**you will get the result for everything**

**appliance on/off**

**live status**

**history**

**via cloud as well as local**

**and this python file we will be executing on boot**

## **5. Mobile - Cloud - Gateway – Control**

### **Live stream of camera**

**Step 1-** User will turn ON the camera device, it will open in hotspot mode, user will provide Wi-Fi credential, camera name, gateway ip address.

**Step 2-** When app connects with gateway again, app will send camera configuration details to gateway, gateway will save its details in its MySQL table control\_box\_configuration.

**Step 3-** Camera device will now open in Wi-Fi mode and connect with wi-fi. Also connect with gateway using ip address.

**Step 4-** Now camera will live stream the video to the gateway.

**Step 5-** Gateway will store the video in a flash drive, also if user want see live stream, user will choose an option in app for live stream of camera.

**Step 6-** Gateway will response with a webpage of live streaming of camera.

**Step 7-** if user request for live stream on cloud, gateway will stream the video to cloud and cloud will stream the video to the user.

**Step 8-** The main advantage of storing and streaming video with gateway is, we can access older recordings, intruder detection, motion detection and face recognition can be done in gateway by using a python program.

## **7. ControlBox – Gateway**

### **Activation of control box to gateway**

**Step 1-** User will give the credential to control box as written in point 2. After this control box will be switched to Wi-Fi mode also app will connect with the wi-fi.



**Step 2- Now app will request to gateway for the webpage to save the new control box configurations.**

**Step 3- If admin user want to create an appliance policy, admin user app will publish a JSON message with topic Gateway/AppliancePolicy**

**JSON message will be**

```
{“UserName”:”xyz”, “AccessKey”:”xyz”, “Location”:”xyz”, “Room”:”xyz”,  
“ControlBox”:”xyz”, “ApplianceName”:”Appliance”, “TriggerTime”:”00:00:00”,  
“EndTime”:”00:00:00”, “PolicyName”:”abcd”}
```

**Gateway will save the policy in the table name Policy\_table**

<b>PolicyClientLocation</b>	<b>Date</b>		<b>Time</b>	<b>RoomControl</b>	<b>Trigger</b>	<b>End</b>
<b>Name</b>				<b>Box</b>	<b>Time</b>	<b>Time</b>

**Step 4- Now gateway will publish a message to the appliance on Trigger time with topic Room/ControlBox (Username will be gateway)**

**JSON message will be-**

```
{ “Username”:”gateway”, “AccessKey”:”xyz”, “Location”:”xyz”  
“ApplianceName”:”Appliance”, “state”:”ON”}
```

**Step 5- Control box will publish an acknowledgement message.**

**Step 6- On End time gateway will again publish a message with topic Room/ControlBox**

**JSON message will be-**

```
{ “Username”:”gateway”, “AccessKey”:”xyz”, “Location”:”xyz”  
“ApplianceName”:”Appliance”, “state”:”OFF”}
```

**Step 7- Control box will again send an acknowledgement message.**

# **ALREADY IMPLEMENTED CLOUD, LIVESTATUS AND HISTORY GETTING ALONG WITH LOCAL CONFIGURATIONS**

## **8. WebCloud – Gateway**

**Get gateway configurations**

**TO GET THE CONFIGURATION DETAILS AND HISTPR**

**Step 1- When user create an account with cloud, cloud will provide him a domain name, user id, password and access key.**

**Step 2- Now user will open app and click on gateway setting, app will request for webpage of gateway setting.**

**Step 3- Gateway will response with a webpage and app will show the webpage.**

**Step 4- Webpage will be secured with a user id, password which was created by user when he was configuring the gateway. User will enter its user id and password.**

**Step 5- In web page there will be an option connect with cloud.**

**Step 6- User will click on this option. It will ask for domain name, user id, password and access key.**

**Step 7- It will try to connect with cloud, If all the credential will be right then gateway will connect with the cloud.**

**Step 8- Now cloud will ask for the gateway configuration and other database files.**

**Step 9- Gateway will provide all the configuration and other information saved in MySQL database to the cloud. Or simply gateway will share the MySQL information with the cloud with a http web page.**

## Get appliance usage history

**Step 1-** In app user will select the appliance for which he wants to get the usage history.

**Step 2-** App will request for a webpage which has access to table

**Room\_ControlBox\_Appliance** in MySQL database.

**Step 3-** Cloud will response with a webpage having option Start date and End date.



Start Date

END Date

**Step 4- User will enter the Start date DD/MM/YYYY and End Date DD/MM/YYYY.**

**Step 5- If cloud has usage history of the appliance it will directly send the information of usage history to the app otherwise it will again generate a request for the usage history with the gateway using http.**

**Step 6- Based on date python will retrieve information from table Room\_ControlBox\_Appliance in MySQL database and send it to cloud using webpage.**

**Step 7- Now cloud will again forward the information to the app using webpage.**

## **Users data**

**Step 1- User will click on users option in the app. App will request for a webpage which has access to table user\_data.**

**Step 2- Cloud will response with a webpage asking for user id and password.**

**Step 3- User will enter the information now again a web page will open with following options :**



**Step 4- User will click on first option i.e. Users.**

**Step 5-** If cloud has user data it will directly send the information to the app using webpage otherwise it will again generate a request for the user data with the gateway using http.

**Step 6-** Now python will retrieve information from table user\_data in MySQL database and send it to cloud using webpage.

**Step 7-** Now cloud will again forward the information to the app using webpage.

## **9. ControlBox - Gateway - Cloud - Mobile**

### **Security Alarm notifications**

**Step 1-** Any alarm like Water tank empty/full alarm, motion detection alarm, plant watering alarm etc. will be published by Topic Alarm/Room/ControlBox with a JSON message.

JSON message will be-

```
{“Appliance”:”WaterTank/MotionDetector....”, “AlarmType”:”Full/Empty/Detected/.....” }
```

**Step 2-** Gateway will receive the message and forward it cloud using mqtt bridge.

**Step 3-** Mobile should subscribe the Alarm topic. If mobile is connected with the gateway using wi-fi the message will reach directly and a notification pop up will show on the screen of mobile and also mobile will start ring.

**Step 4-** If mobile is connected with the cloud, the mqtt bridge will publish the message to the cloud, cloud broker will forward the message to the mobile.

**Step 5-** In case of motion detection alarm by camera, if an image or a short video clip (10-20 sec.) is to be sent via message, Camera will stream the image or short video clip to gateway using http. Gateway will save the image/video and forward it to the cloud using http.

**Step 6-** Cloud will send a request to the user app to download the image or video using webpage.

**Step 7-** App will download the image/video and inform the user about the alar

